Carte tempo – générateur de portes et retards

Table des matières :

Table des figures :

1 Description

La carte **tempo** pour fonction de générer les différentes impulsions nécessaires au séquencement d'une acquisition. Avec un JITTER faible.

Elle est au format VME et a une largeur d'une unité.

L'originalité de cette carte est de permettre la génération d'impulsions de retards et largeurs variable tout en gérant le temps mort. Elle a aussi des fonctionnalités permettant :

- le comptage du nombre total d'événements ;
- le comptage des événements rejetés ;
- La production d'une interruption après chaque événement accepté (délai réglable) ;
- La possibilité de faire stopper une acquisition après un temps écoulé ou un nombre d'événement arrivé ;

Pour des facilités de mise au point d'une acquisition, un générateur d'événement manuel est mis à disposition.

1.1 Relation entre les signaux

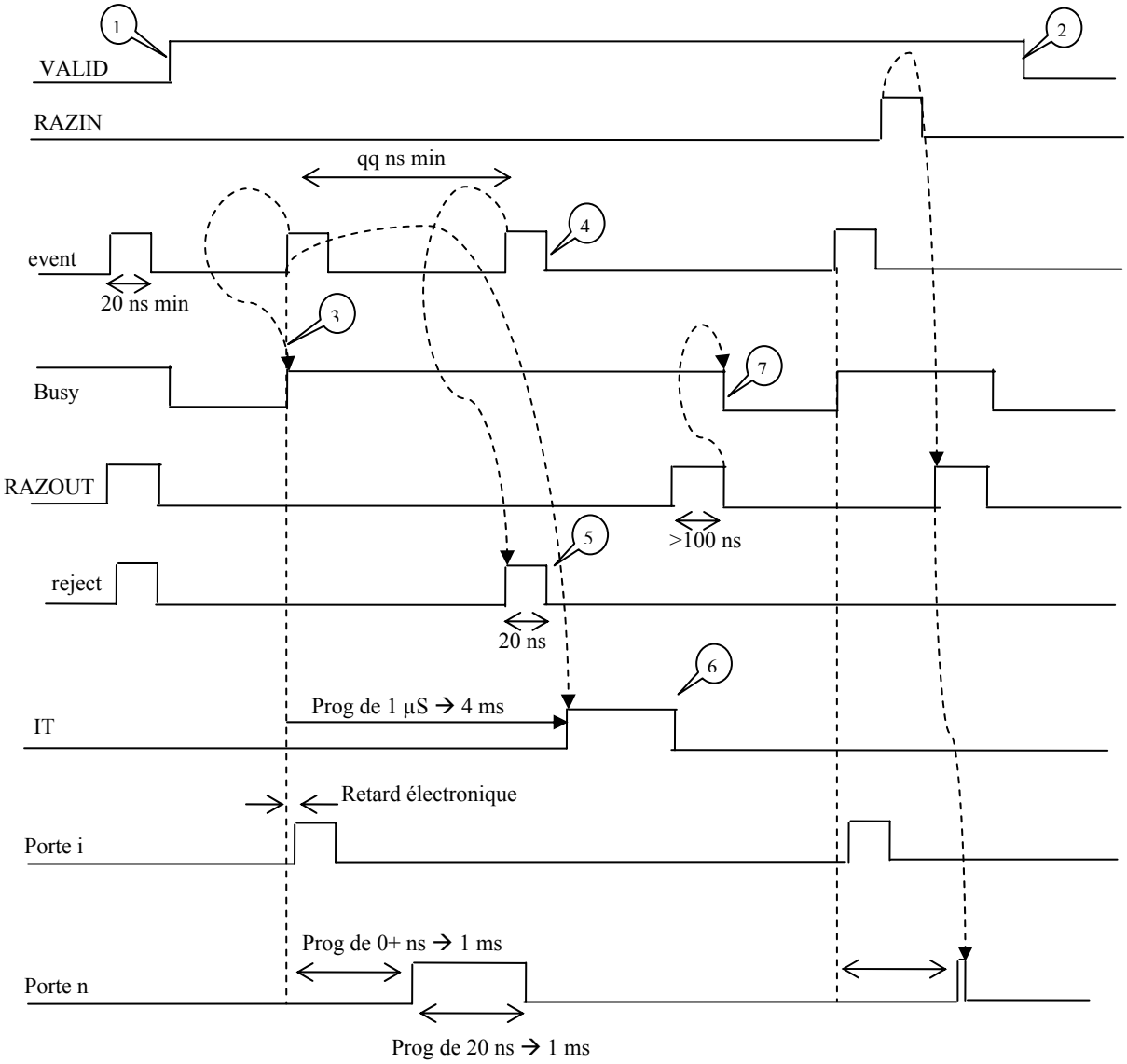
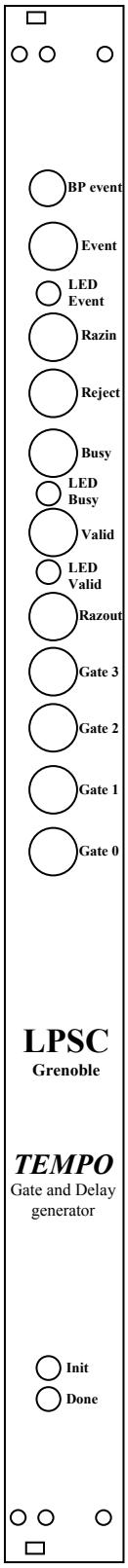


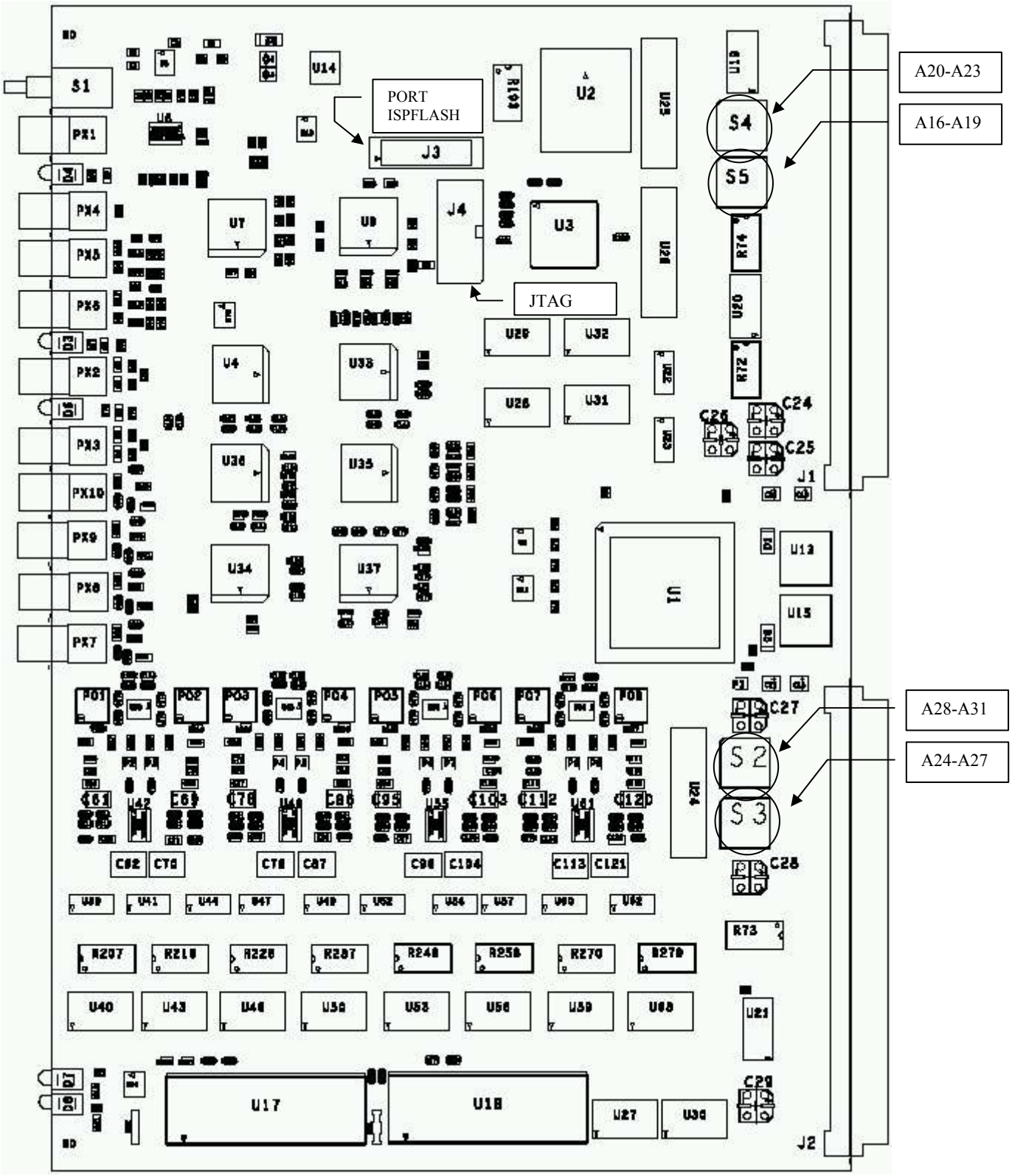
Figure 1 : Relation entre les signaux

- (1), (2) : *La carte est activée par une écriture VME, alors VALID passe à l'état actif et la carte sort du temps mort. Si elle est désactivée, la carte arrête de générer des portes* ;
- (3) : *Un événement est accepté uniquement si la carte n'est pas en temps mort ;*
- (4), (5) : *Si un événement arrive alors que la carte est en temps mort un signal **reject** est généré, il n'y a pas de contrainte de rapidité entre la réception de **EVENT** et la génération de **reject** ;*
- (6) : *Le fait de recevoir un **EVENT** valide génère une interruption après un délai programmable ;*
- (3), (7) : *La carte entre en **Busy** lorsque un **EVENT** valable est reçu, et en ressort dès que le signal **RAZout** est relâché. L'entrée en temps mort doit être la plus rapide possible ;*
- *Le signal **RAZout** est généré soit si une écriture VME dans le registre approprié a eu lieu et si les **gates** sont terminées, soit par un RAZin valide (c'est-à-dire pas entre IT et écriture VME) et dans ce cas, il doit arrêter toutes les portes potentiellement encore en route, et arrêter la génération d'IT ;*
- *Le signal **RAZout** peut être rallongé si nécessaire. Ainsi lorsque un **RAZin** arrive, on rallonge le **RAZout** pour permettre la décharge des capas. Dans le cas où il arrive après, on rallonge uniquement si nécessaire (à savoir si les décharges ne peuvent s'imbriquer).*

1.2 Façade avant



1.3 Position des composants



2 Spécifications

2.1 Entrées

Tous les signaux sont au format NIM et de polarité positive.

- **Event** : Il signale l'arrivée d'un événement ;
- **RAZin** : Ce signal permet de sortir du temps mort et de revalider l'acquisition si l'IT n'a pas encore été générée. Il doit durer au moins 100 ns.

2.2 Sorties

Tous les signaux sont au format NIM et de polarité positive.

- **Valid** : Il signale le fait que la carte est activée et que les événements peuvent être acceptés.
- **Busy** : Il fonctionne en conjonction avec **EVENT** et **RAZ**. Il est actif dès la réception d'un événement, il est libéré en même temps que **RAZ**. En dehors de la validation, la carte est en **Busy** .
- **RAZout** : Il est activé juste avant la sortie du **Busy** par une écriture VME. Il dure au minimum 100 ns.
- **Reject** : Il est activé lorsqu'un événement arrive et que la carte tempo est en **Busy**.
- **Gates** : Signaux activés après un retard programmable (par rapport à la réception d'un signal **EVENT** valide, c'est-à-dire lorsque la carte n'est pas en temps mort). La durée de l'activation sera elle aussi programmable

2.3 Interruption

Signal de sortie VME. Une interruption doit être générée après qu'un délai se soit écoulé par rapport à la réception d'un signal **EVENT** valide. Ce retard est programmable par pas de 62,5 ns et pour une durée maximum de 4,096 ms.

2.4 Consommation

+5V : 2,5 A
-12V : 240 mA

2.5 Temporelles

Au niveau du jitter, il est à noter qu'il faut prendre des précautions lors du réglage de l'arrivée de l'interruption porte et des différentes portes. Si il y a corrélation un bruit supplémentaire équivalent à ~2% de la pleine gamme s'ajoute. Dans le cas d'une corrélation entre portes, le problème est légèrement différent, dans la zone de corrélation, les 2 portes auront tendance à commuter au même moment malgré une modification substantielle des valeurs DAC.

Largeur d'un **Event** : > 6 ns ;
Délai entre **Event** et **Busy** : <4 ns ;
Délai entre **Event** et **Reject** : < 4ns ;
Délai entre fin **RazOut** et sortie **Busy** : <1 ns
Stabilité des réglages (en dehors de la zone de recouvrement du bas de calibre) : < 2% ;

3 Solution technique

Pour pouvoir fonctionner de manière asynchrone, et ce de manière la plus rapide possible. Le design sera basé sur de l'électronique PECL et sur des charges de condensateur à courant constant.

Lorsqu'un condensateur est chargé à courant constant, la valeur de la tension à ses bornes est :

$$i = C\frac{\Delta U}{\Delta t}$$

Avec $i = \frac{dq}{dt} = \frac{\Delta q}{\Delta t}$ et $Q = CU$ on obtient :

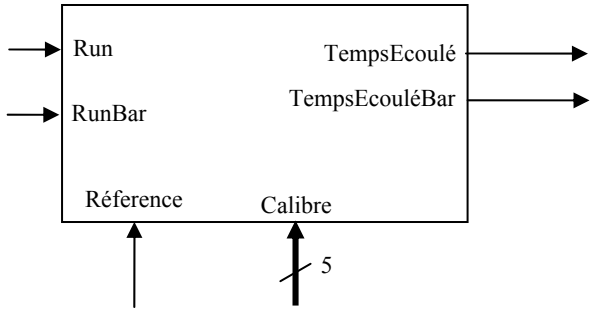
$$\Delta U = \frac{i}{C} \bullet \Delta t$$

3.1 La Base de temps

Elle est basée sur 3 modules :

- Un commutateur de courant;
- Un module de condensateurs ;
- Un comparateur ultra-rapide avec référence programmable ;

3.1.1 Interface



- **Run, RunBar** : contrôle l'activation de la base de temps, doit être maintenu pour faire fonctionner la base de temps ;
- **TempsEcoulé, TempsEcouléBar** : passe à l'état haut lorsque le délai est écoulé ;
- **Calibre** : Permet de choisir la gamme de temps désirée ;

3.1.2 Le commutateur de courant

Il sera tout simplement basé sur une paire différentielle commandée par des signaux PECL différentiels. La référence de courant est fournie par un miroir qu'il est possible d'ajuster grâce au un potentiomètre multitour et à la mesure de tension sur la résistance de 1kΩ.

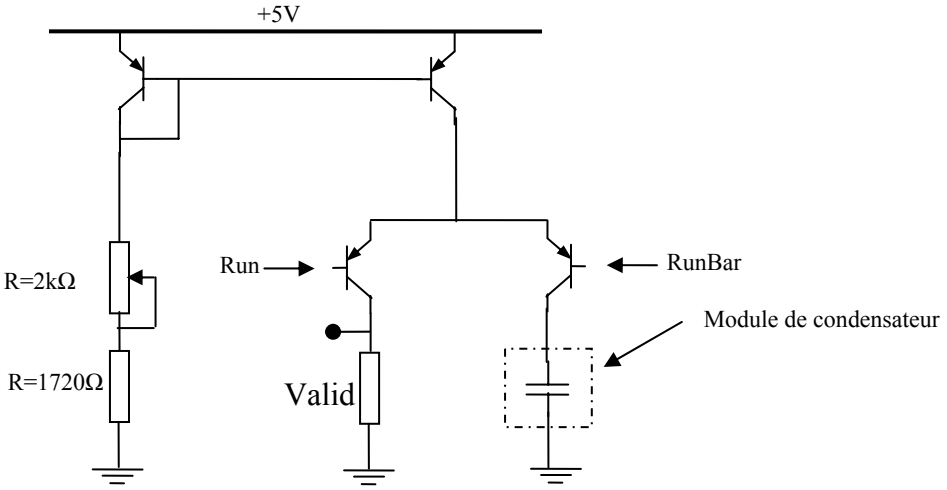


Figure 2 : commutateur de courant

3.1.3 Le module de condensateur

Ce module contient les condensateurs de référence pour la base de temps, ce sont des condensateurs précis et à faible dérive dans le temps et en température. Il contient aussi les interrupteurs de sélection de gamme et de décharge rapide des condensateurs.
Les différentes gammes de temps visées et leur condensateur associé sont (pour un courant de charge de 2,5 mA et une tension maximum de 2,5 Volt) :

gamme	Condensateur	Pas de réglage (1/100ème)
100 ns	100 pF	1 ns
1 µs	1 nF	10 ns
10 µs	10 nF	100 ns
100 µs	100 nF	1 µs
1 ms	1 µF	10 µs

De plus pour permettre une décharge rapide des condensateurs (<20µs), il convient de mettre autant de switches que possible en parallèle (le courant DC max dans le switch est 128 mA).

Le problème de délai se pose uniquement pour les 2 dernières gammes à savoir 100µs et 1 ms. Pour que 5*RC<20µs, il faudrait $R_{100\mu s} < \frac{20\mu s}{5*C} = 40\Omega$ de même $R_{1ms}<4\Omega$. Une décharge à 5*RC correspond à une décharge à 99,32%. Pour des raisons pratiques $R_{1ms}\approx 5{,}5\Omega$ soit 5*RC≈27,6µs.

Attention, dans le premier calibre, l'incertitude dû aux capacités parasites est assez élevé, de l'ordre de 10 %. Dans le calibre 1 ms, il convient d'avoir un signal RAZ suffisamment long pour interdire tout redéclenchement lors de la décharge du condensateur.

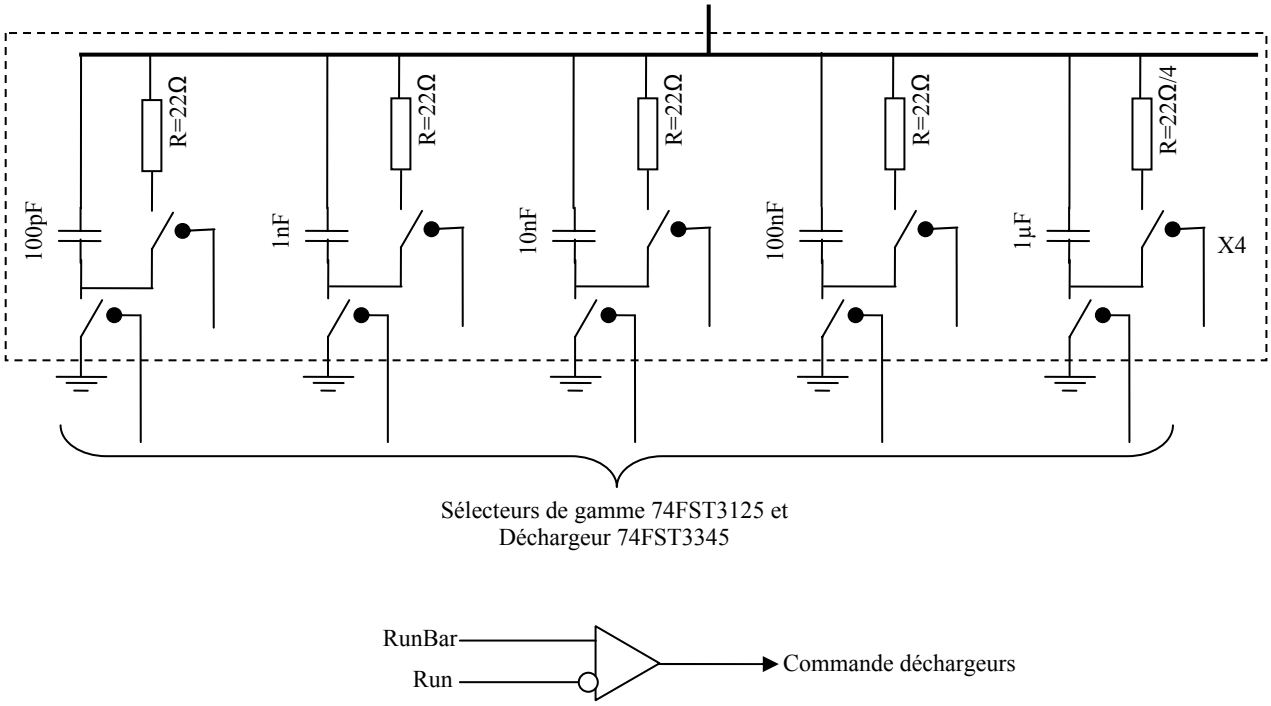


Figure 3 : module de condensateur sélectionnable

3.1.4 Le comparateur ultra-rapide à référence programmable

Le comparateur utilisé est un MAX9601 de chez MAXIM, il permet de faire des comparaisons avec un temps de propagation de 500 ps (typ) tout en fournissant un signal de sortie PECL complémentaire. L'idée est de positionner l'une des entrées du comparateur à un potentiel préréglé par un DAC programmable et de faire la comparaison avec le potentiel aux bornes du module de condensateur chargé à courant constant.
Le DAC utilisé sera sur 12 bits, sachant que la pleine gamme sera divisée en 100. C'est pour éviter une incertitude trop grande. Avec 8 bits on aura eu +/-40% sur le pas, ici on aura +/-2,5%.

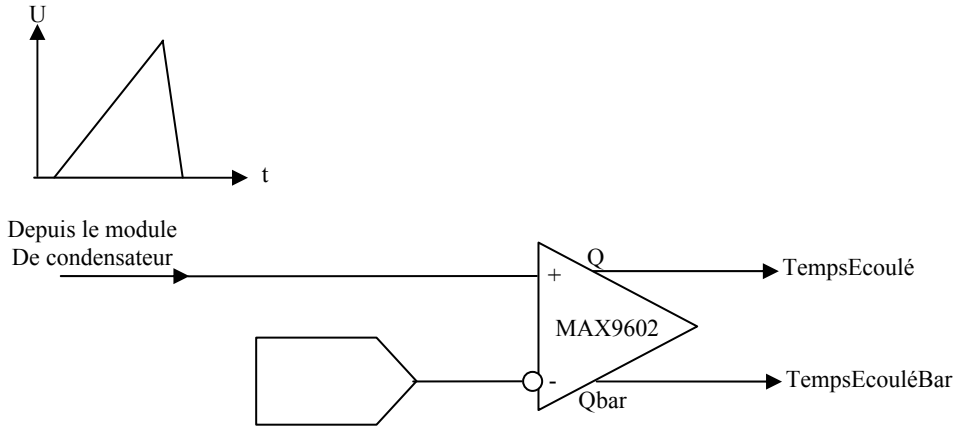


Figure 4 : comparaison

3.1.5 Bilan temporel de la base de temps

Le chemin critique pour l'activation de la base de temps est celui dû à la désactivation du déchargeur. C'est-à-dire qu'il faut compter :

- le tco (Time of Clock to Output) de la bascule (=410 ps max pour un fanout de 1) ;

- le temps de désactivation du chargeur (8 ns max) ;
- Le temps de traversée du comparateur (700 ps max) ;
- Le temps de traversée de la porte OR (550 ps max) ;
- Le temps de réaction de la bascule au reset (470 ps max) ;

Soit un temps incompressible max de 10,2 ns. Ce sera la durée minimale du monostable. Il convient d'ajouter ce délai dans le budget du retard programmable et de la largeur programmable.

3.2 Conversion de l'entrée NIM en PECL

Pour faire cette opération, il faut aussi utiliser un comparateur ultra-rapide.

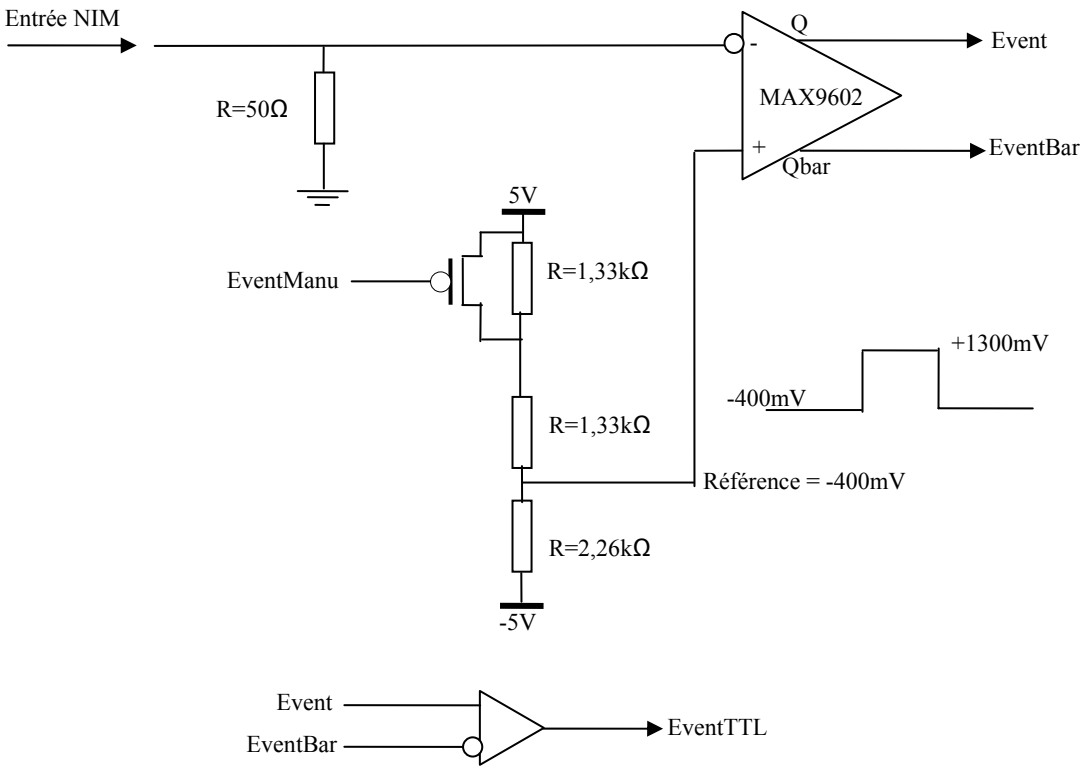


Figure 5 : conversion NIM → PECL

Un générateur d'événements manuel est ajouté à la référence pour ne pas ajouter de capacités parasites sur l'entrée NIM et donc ralentir le système.

3.3 Conversion des sorties PECL en NIM

Pour assurer la conversion PECL vers NIM, il faut utiliser le montage suivant :

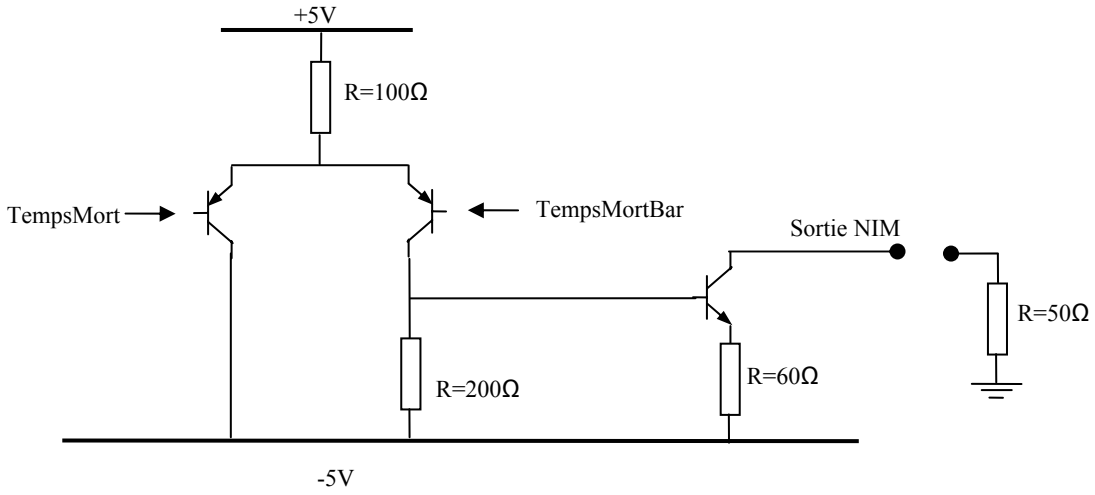


Figure 6 : Conversion PECL diff → NIM

3.4 La gestion du temps mort et du RazOut

La carte tempo doit être en temps mort en dehors de la validation de la carte et lorsqu'un événement valide a été accepté.
La solution est d'utiliser une bascule PECL MC100EP31 pour mémoriser cela. Cette partie fonctionne en logique négative.

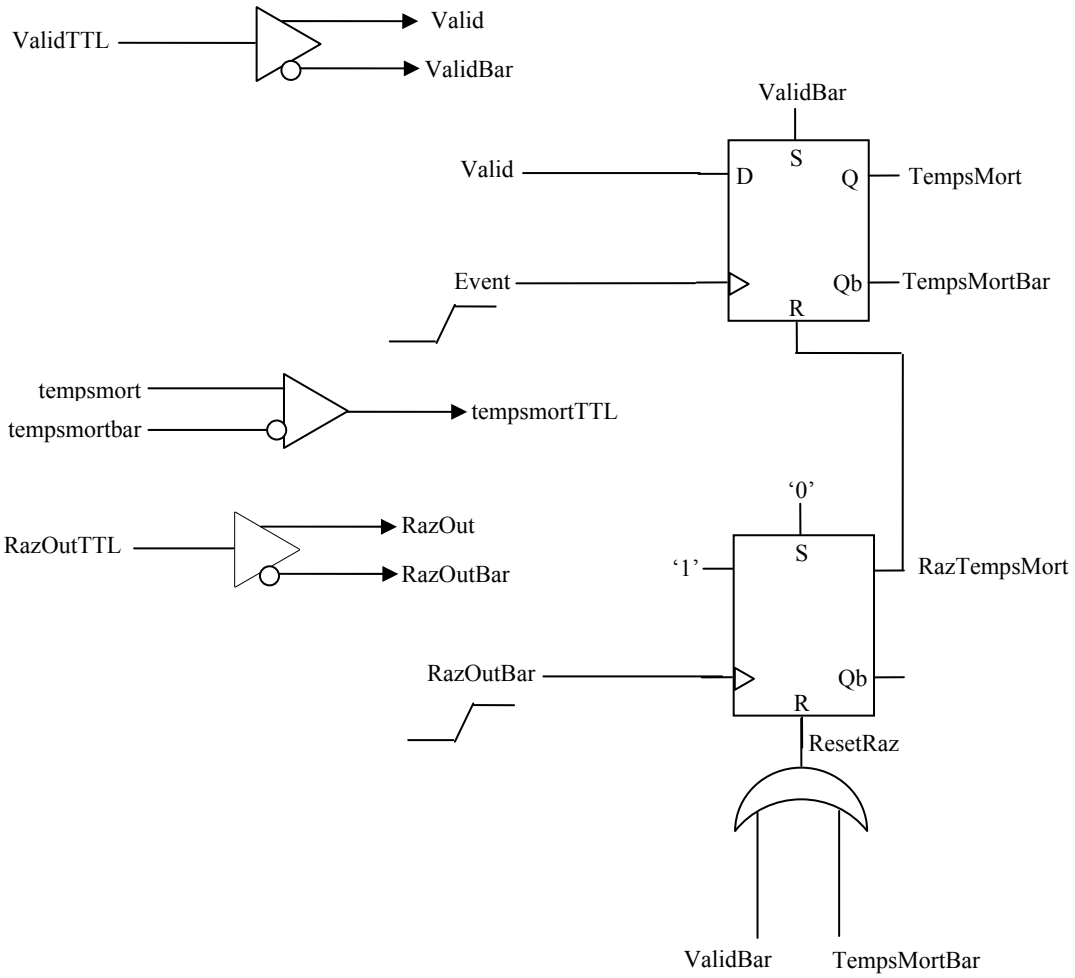


Figure 7 : circuit du temps mort et de RAZ

Quand la carte est en temps mort, **TempsMort**='1' et **TempsMortBar**='0'. Elle ne peut passer dans cet état que quand **Valid** vaut '1' et qu'il y a un **Event**. Si **RazOutBar** ou **Valid** valent '0' alors la carte passe en temps mort.
Pour sortir du temps mort, il faut toujours faire un **RazOut**, que soit après un événement ou après la validation.

Chronogramme de l'activation de la carte :

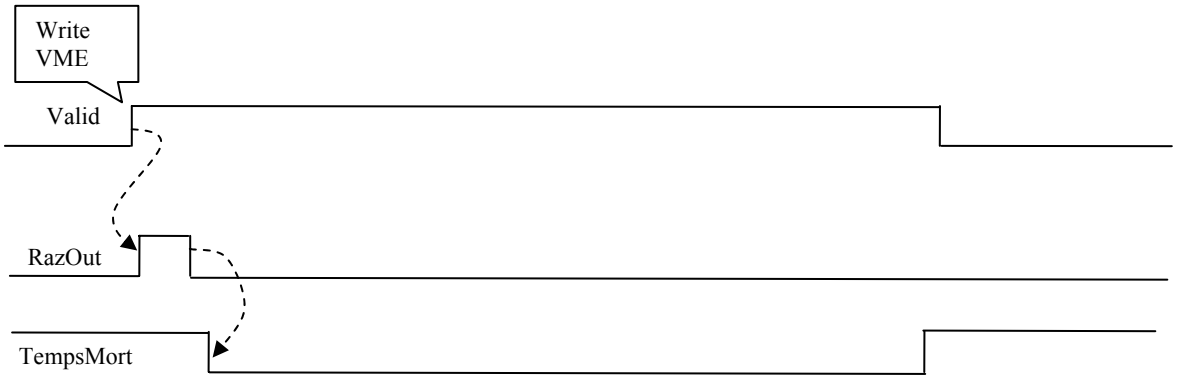


Figure 8 : Chronogramme de validation

Avec cette solution, le délai typique entre un événement et une sortie temps mort est environ :
T_{typ} = 500ps (comparateur) + 400ps (bascule) + 1000 ps (PECL→NIM)=1900 ps

Pour ce qui est du **ResetRaz** et du **RazTempsMort**, l'impulsion min sera due à :

- t_{sr} bascule RAZ =300 ps min ;
- t_{sr} bascule TempsMort =300 ps min ;
- Tpd porte OU = 200 ps min ;

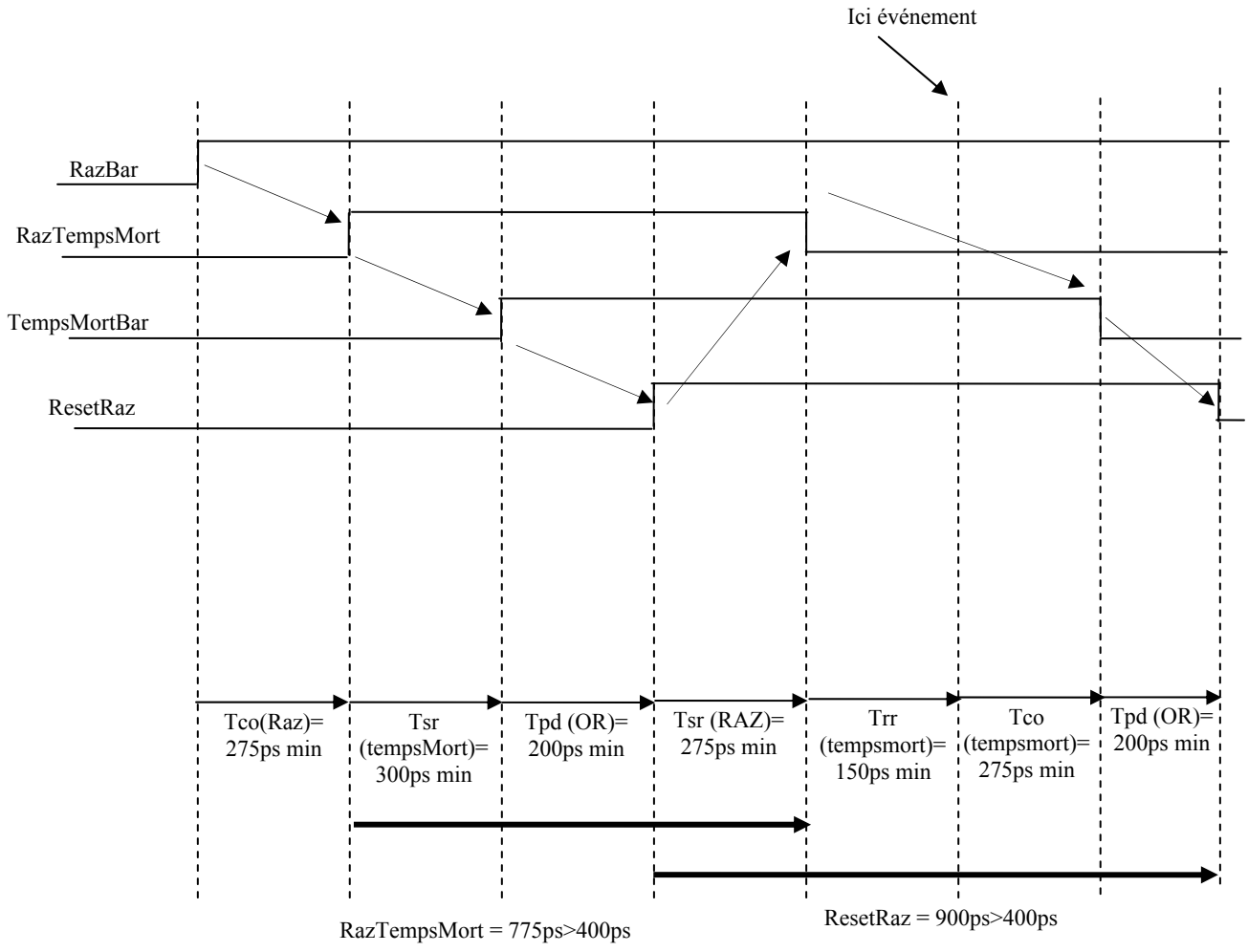


Figure 9 : Chronogramme de durée min des reset

3.5 La gestion des portes

Le montage est basé sur 2 bascules et 2 bases de temps, permettant de réaliser d'une part le délai entre le front de l'événement et la sortie de l'impulsion, et d'autre part la largeur de l'impulsion.

La première bascule passe à '1' en sortie si la carte n'est pas en temps mort OU si elle déjà activée (elle est alors maintenue à '1'). Pour repasser à '0', il faut que le délai spécifiant le retard soit écoulé OU que l'on Remette à Zéro le système.

La deuxième bascule gère la largeur de la porte, mais aussi, elle peut être activée plus rapidement grâce au multiplexeur en entrée. Ainsi si l'on souhaite un délai très proche de 0, on aura une valeur de environ 3 à 4 ns de bout en bout.

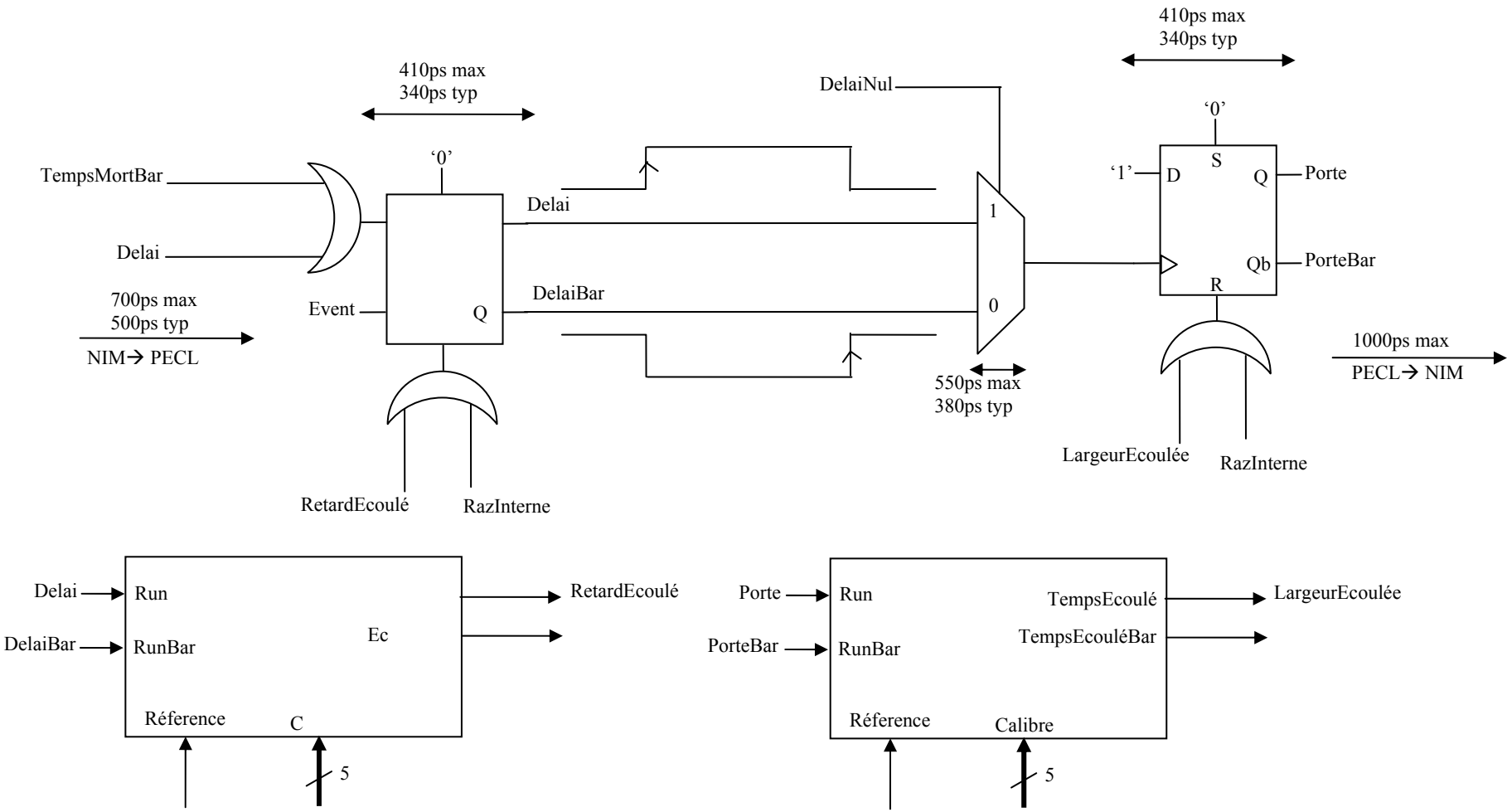


Figure 10 : gestion d'une porte

Tous les temps Indiqués, le sont pour un fanout de 1. Ne connaissant pas les capas d'entrée, des divers composant PECL, l'allongement de temps de propagation ne peut être déterminé avec précision (Au pire un facteur 2).

3.6 Gestion du RazInterne

Pour avoir le temps de décharger les condensateurs de forte valeur lors du de la réception d'un **RazIn**, le **RazOut** et **RazInterne** sont égaux et rallongés d'une durée programmable qui doit être suffisante à la décharge des condensateurs
Dans le cas ou le système va jusqu'à l'IT, un **RazInterne** est généré immédiatement alors que le **RazOut** ne l'est pas, lui est généré sur l'écriture VME. Ainsi le temps de décharge est partiellement masqué par le temps de calcul du processeur et **RazOut** n'est prolongé que de ce qui est nécessaire.

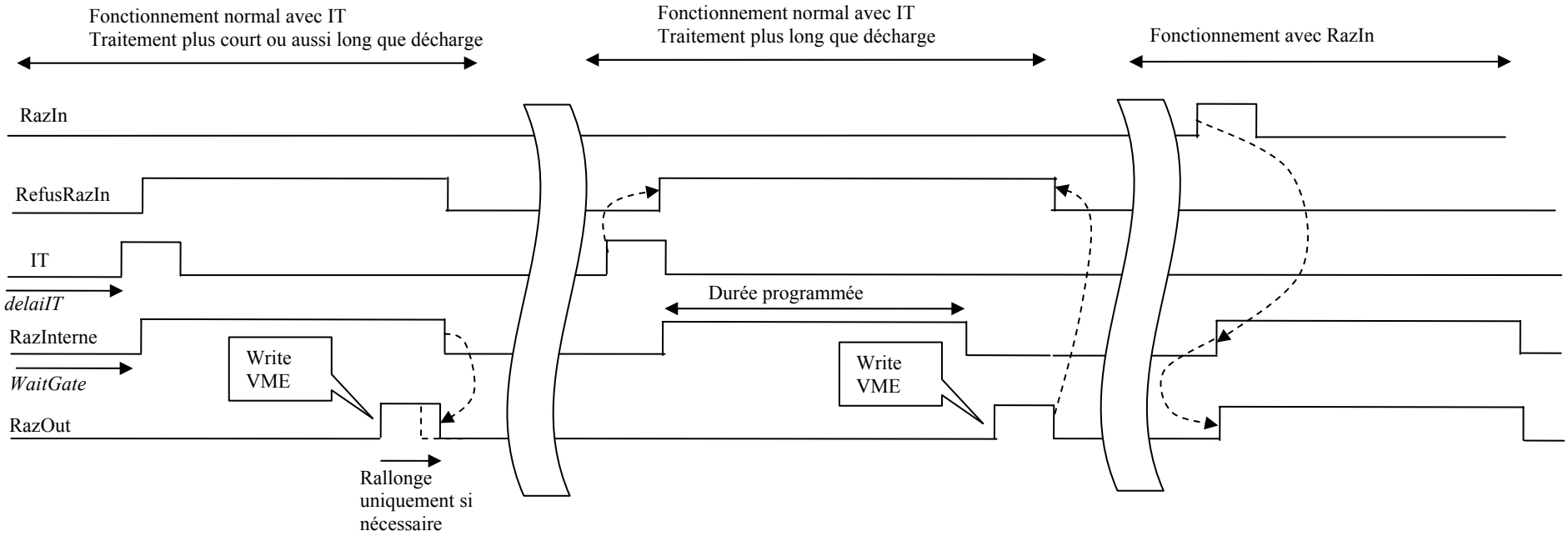


Figure 11 : Chronogramme de gestion du RazInterne dans le cas de RazIn

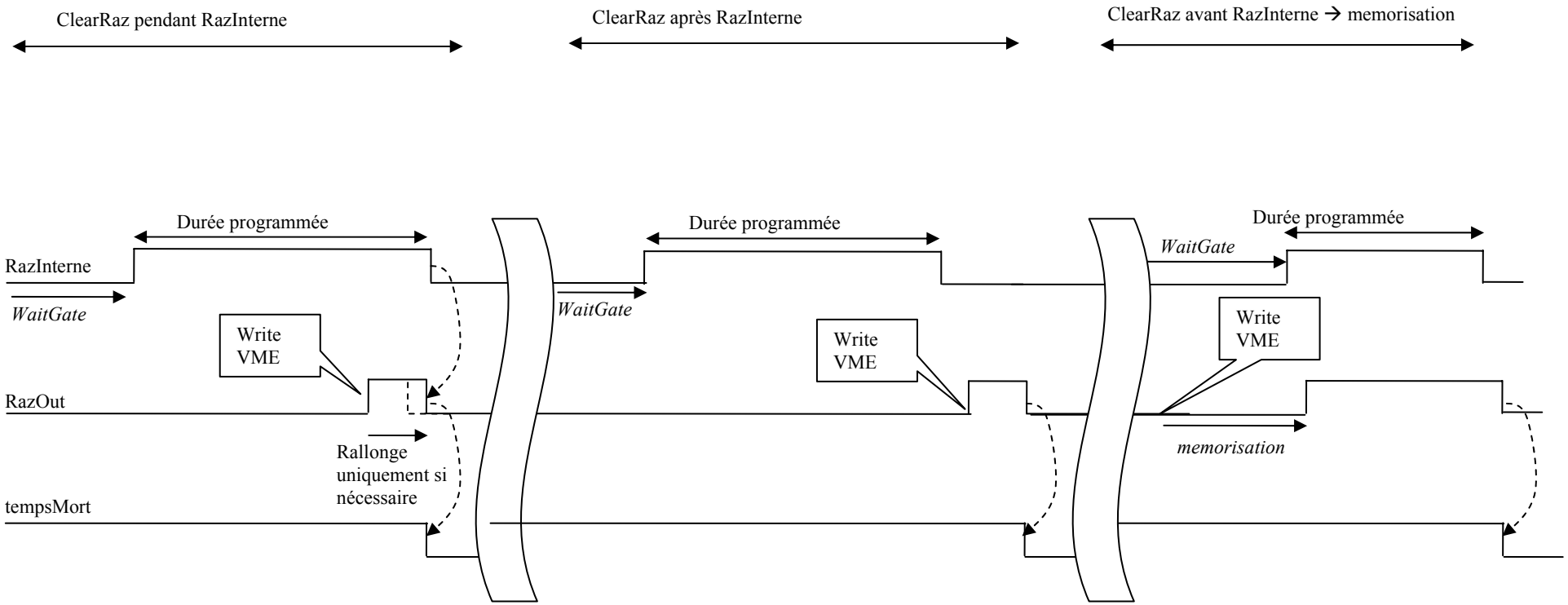


Figure 12 : Chronogramme de gestion du RazInterne dans le cas de ClearRaz

3.7 Génération du signal Reject

Il faut générer un signal chaque fois qu'un nouvel événement arrive et que l'on est en temps mort.

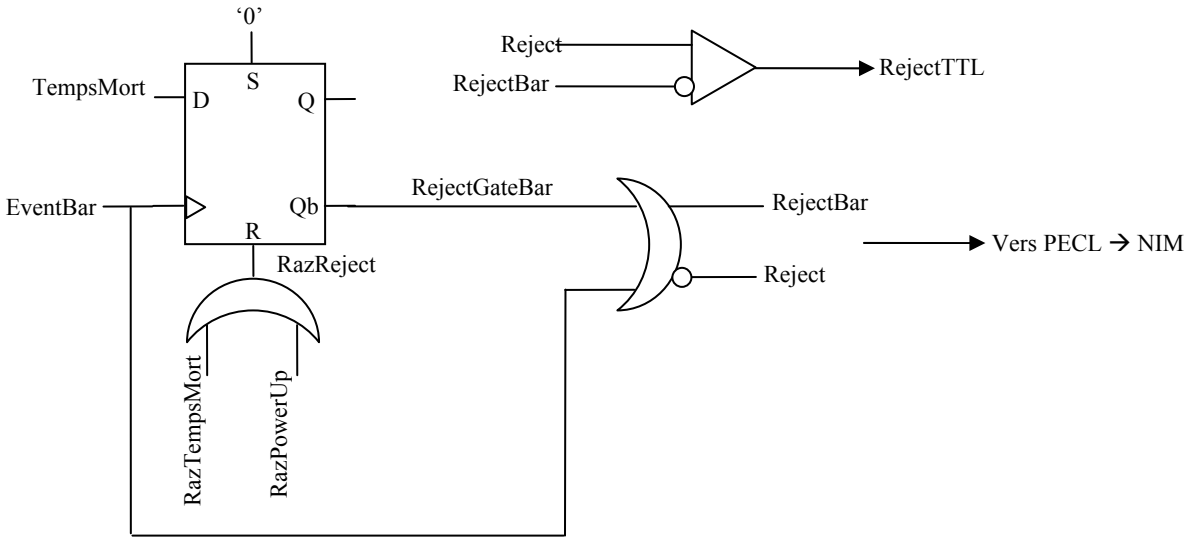


Figure 13 : principe de génération de reject

Le chronogramme de fonctionnement de cette partie de design est :

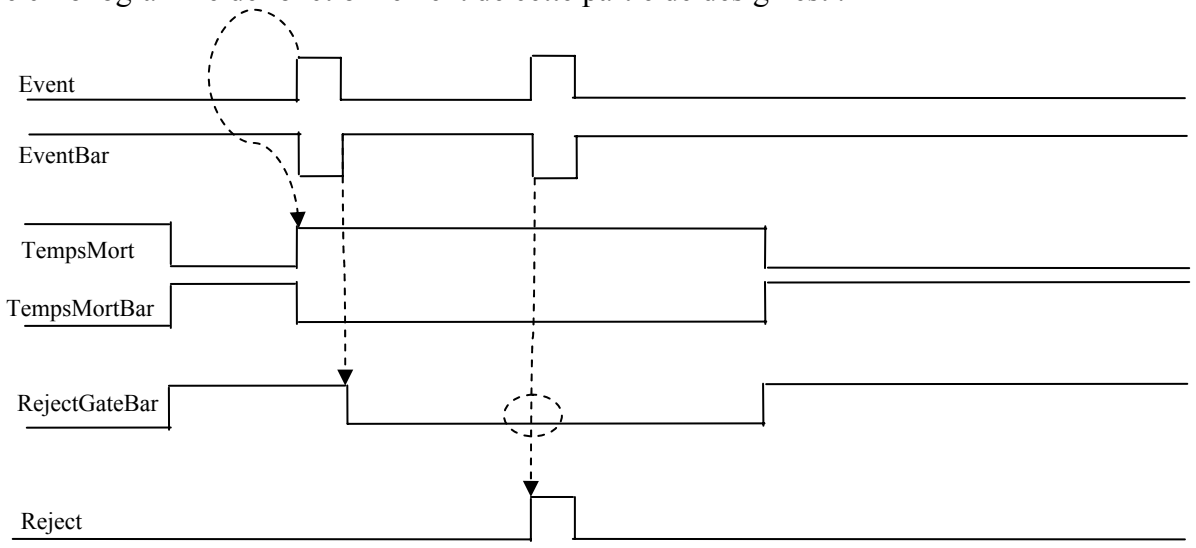


Figure 14 : Chronogramme de reject

Le signal RejetGateBar sert à éviter que l'évènement valide soit comptabilisé comme un évènement rejeté. Ainsi dès qu'un évènement arrive alors que l'on n'est pas en temps mort, le signal rejectGateBar n'est pas actif, mais s'active sur le front montant signalant la fin de l'évènement ainsi les évènements suivants seront véritablement comptabilisés. Le signal RazPowerUp sert à faire démarrer la bascule sur un état déterminé, à savoir tout événement arrivant après la mise sous tension est un évènement rejeté, (TempsMort est actif à la mise sous tension).

3.8 EPLD d'interface VME : Tempo_VME

3.8.1 Adressage

3.8.1.1 IDCODE (read only 32)

Situé à l'adresse de base, permet de voir si la carte est bien présente
Base + 0 soit $(000000)_{bin}$ → $(00)_{hex}$

IDcode (32 bits)		
31	...	0

La valeur attendue à la lecture est $(BEBEDADA)_{hex}$.

3.8.1.2 Validation (write only 32)

Une écriture à cette adresse permet de valider l'acquisition et de choisir si l'on veut activer le contrôle d'acquisition temporel et numéraire. Dans l'un de ces cas, si le temps écoulé ou le nombre d'événements est atteint, l'acquisition est stoppée. Les 2 types de contrôle d'acquisition peuvent être sélectionnés simultanément, c'est à la première échéance arrivée que l'acquisition s'arrête.
Si le contrôle d'acquisition est activé, l'échéance de celle-ci est signalé par une interruption de niveau programmable voir (3.8.1.11 page 23).
Après l'interruption, si l'on veut redéclencher ce type d'évènement il faut réecrire dans ce registre pour tout d'abord arrêter effectivement l'acquisition (valid='0') puis revalider le tout.
Base + 2 soit $(000010)_{bin}$ (soit pour sur 32 bits base + 4) → $(04)_{hex}$

Durée RAZ (10 bits)			Contrôle numéraire	Contrôle temporel	Validation (1 bits)
12	…	3	2	1	0

La durée est réglable par pas de 62,5 ns → 64 µs.

Dans ce registre on écrit aussi la durée minimum de décharge des condensateurs (Durée RAZ), c'est-à-dire en fonction du calibre. Les valeurs à coder sont :

gamme	**Condensateur**	**Valeur du circuit résistif (max)**	**Temps (5*RC)**	**Valeur pour T=62,5 ns**
100 ns	100 pF	29Ω	12ns	1
1 µs	1 nF	29Ω	120ns	>2
10 µs	10 nF	29Ω	1,2 µs	>20
100 µs	100 nF	29Ω	12 µs	>200
1 ms	1 µF	7,25Ω	36,25µs	>580

3.8.1.3 DAC_CTRL (write only 32)

Une écriture à ces 2 adresses permet de configurer les DAC et donc les différents temps programmables en conjonction avec le sélecteur de gamme.
Base + 4 + numéro de quart

- $(000100)_{bin}$: Porte 0 et 1 ; (soit pour sur 32 bits base + 8) → $(08)_{hex}$
- $(000110)_{bin}$: Porte 2 et 3 ; (soit pour sur 32 bits base + 12) → $(0C)_{hex}$

valeur (12 bits)			ADD (2 bits)	
29	…	18	17	16

Sur la carte, il y a physiquement 2 composants DAC implantés, ceux-ci à leur tour contiennent 4 sous-dac. C'est pour cela que pour adresser un DAC il y a une adresse VME $(08)_{hex}$ ou $(0C)_{hex}$ et une sous adresse contenue dans la donnée.
Par exemple, pour programmer les 2 DAC de la voie 0, il faut utiliser l'adresse $(08)_{hex}$ et faire successivement 2 écritures avec ADD à « 00 » et « 01 ».
Mapping des DAC :

cible	Adresse	Registre DAC (17..16)
Délai_0	Base+8	00
Largeur_0		01
Délai_1		10
Largeur_1		11
Délai_2	Base+12	00
Largeur_2		01
Délai_3		10
Largeur_3		11

Tableau 1 : Adressage des DAC

Voici quelques exemples de données à transférer suivant le DAC désiré :

		REF0- REF2-	REF0+ REF2+	REF1- REF3-	REF1+ REF3+	
tension DAC	valeur DAC	sous-DAC=0	sous-DAC=1	sous-DAC=2	sous-DAC=3	Pourcentage de la gamme
0,00	0	0000	0001	0002	0003	0
	163	028C	028D	028E	028F	4
0,20	327	051C	051D	051E	051F	8
0,30	491	07AC	07AD	07AE	07A	12
0,40	655	0A3C	0A3D	0A3E	0A3F	16
0,50	819	0CCC	0CCD	0CCE	0CCF	20
0,60	983	0F5C	0F5D	0F5E	0F5F	24
0,70	1146	11E8	11E9	11EA	11EB	28
0,80	1310	1478	1479	147A	147B	32
0,90	1474	1708	1709	170A	170B	36
1,00	1638	1998	1999	199A	199B	40
1,10	1802	1C28		1C2A		44
1,20	1966	1EB8	1EB9	1EBA	1EBB	48
1,30	2	2144	2	2146		52
1,40	2293	23D4	23D5	23D6	23D7	56
1,50	2457	2664	2	2666		60
1,60	2621	28F4	28F5	28F6	28F7	64
1,70	2	2B84		2B86		68
1,80	2949	2E14	2E15	2E16	2E17	72
1,90	3112	30A0	30A1	30A2	30A3	76
2,00	3276	3330	3331	3332	3333	80
2,10	3440	35C0	35C1	35C2	35C3	84
2,20	3604	3850	3851	3852	3853	88
2,30	3768	3AE0	3AE1	3AE2	3AE3	92
2,40	3932	3D70	3D71	3D72	3D73	96
2,50	4096	4000	4001	4002	4003	100

Tableau 2 : tableau de valeurs DAC

Il faut charger une valeur différente dans les DAC en fonction des modes d'acquisition désirés. L'équation du DAC étant $Vout = \left(\frac{2,5 \bullet N}{4096}\right)$ soit $N = \frac{Vout \bullet 4096}{2,5}$

3.8.1.4 RANGE_CTRL (write only 32)

- $(001000)_{bin}$: Porte 0/1 ; (soit pour sur 32 bits base + 16) → $(10)_{hex}$
- $(001010)_{bin}$: Porte 1/3 ; (soit pour sur 32 bits base + 20) → $(14)_{hex}$

Permet de sélectionner le calibre sur les modules de condensateur, ainsi que de piloter le sélecteur délai nul, pour ignorer la base de temps de contrôle du délai. L'écriture affecte 2 voies simultanément. Voie 0 et 1 puis Voie 2 et 3. Cela fait l'écriture dans deux 74LVC16374.

Gamme Délai 1/3 (5 bits)			Gamme Largeur 1/3 (5 bits)			Délai nul 1/3 (1 bits)	Gamme Délai 0/2(5 bits)			Gamme Largeur 0/2 (5 bits)			Délai nul 0/2 (1 bits)
21	…	17	16	…	12	11	10	…	6	5	…	1	0

Valeur binaire (5 bits)	Gamme (largeur ou Délai)
00001	100 ns
00010	1 µs
00100	10 µs
01000	100 µs
10000	1 ms
autres	invalides

Si le **delai_nul** est sélectionné, ou si une voie n'est pas utilisée il faut tout de même programmer la gamme au minimum c'est-à-dire à 100 ns et régler la valeur DAC à 1024 pour des raisons de stabilité. Lors de la modification de la gamme et/ou des valeurs DAC, il conseillé d'envoyer un clear_raz voir 3.8.1.5 page 22.

3.8.1.5 Clear RAZ (write only 32)

Une écriture à cette adresse permet de générer un signal RazOut et de sortir du temps mort. Voir Figure 11, page 16 et Figure 12, page 17.

$(001100)_{bin}$: (soit pour sur 32 bits base + 24) → $(18)_{hex}$

3.8.1.6 Programmation temporelle (write only 32)

Si l'on souhaite une acquisition contrôlée par le temps (avec une résolution de 32µs pour aller jusqu'à 36 heures), la résolution effective est de 32µs), il faut programmer la valeur souhaitée à cette adresse.

$(001110)_{bin}$: (soit pour sur 32 bits base + 28) → $(1C)_{hex}$

TimeCode (32 bits)		
31	...	0

Budget ressource : 32 bascules + 9 bascules =41.
Le compteur est mis à zéro à la fin de la validation.

3.8.1.7 Lecture du nombre d'événement rejetés (read only 32)

$(010000)_{bin}$: (soit pour sur 32 bits base + 32) → $(20)_{hex}$

Rejected Event (32 bits)		
31	...	0

Ce compteur peut être ré initialisé grâce à une écriture, voir 3.8.1.9.

3.8.1.8 Lecture du nombre d'événement (read only 32)

$(010010)_{bin}$: (soit pour sur 32 bits base + 36) → $(24)_{hex}$

EventCount (32 bits)		
31	...	0

Ce compteur peut être ré initialisé grâce à une écriture, voir 3.8.1.9.

3.8.1.9 Préchargement/reset du nombre d'événement (write only 32)

$(010100)_{bin}$: (soit pour sur 32 bits base + 40) → $(28)_{hex}$

Preload value (32 bits)		
31	...	0

Attention, une écriture à cette adresse efface le compteur d'événements et le compteur d'événements rejetés.
Il est initialisé à $(FFFFFFFF)_{16}$. Lors de l'écriture dans ce registre, il faut se rappeler que le compteur d'événement saturera à la valeur programmée dans ce registre.

3.8.1.10 Programmation de l'interruption évènement (write only 32)

Une écriture dans ce registre permet de sélectionner le niveau de l'interruption, le délai entre l'arrivée d'un événement valide et de revalider l'IT après un cycle d'interruption. Le simple fait d'accéder ce registre valide (ou revalide) l'interruption.

$(010110)_{bin}$: (soit pour sur 32 bits base + 44) → $(2C)_{hex}$

Status_Event			Délai entre événement et IT(16 bits)			Niveau d'IT Porte (3 bits)		
26	…	19	18	…	3	2	…	0

Le retard est réglable par pas de 62,5 ns → 4,096 ms.
Attention : lors de la programmation de délai_IT, il faut éviter d'avoir une interruption en même temps qu'une fin de porte ou une fin de retard. Les mesures ont montré un jitter supérieur si cela est le cas.

3.8.1.11 Programmation de l'interruption DAQ_CTRL (write only 32)

Une écriture dans ce registre permet de sélectionner le niveau d'interruption pour le contrôle d'acquisition ainsi que son mot de status. Après une interruption, il faut réécrire dans ce registre pour revalider l'interruption.

$(011000)_{bin}$: (soit pour sur 32 bits base + 48) → $(30)_{hex}$

Status DAQ_CTRL			Niveau d'IT DAQ_CTRL (3 bits)		
10	…	3	2	…	0

Attention, les 2 sources d'interruptions sont chaînés. Le premier interrupter sera l'IT porte, quel que soit son niveau de priorité et ensuite l'IT DAQ_CTRL. Donc en cas d'IT simultanées et de même niveau d'interruption, la premier status retourné est celui correspondant à l'IT porte.

3.8.2 Algorithmes de référence

L'algorithme suivant détaille le séquencement des opérations d'écriture et de lecture pour une utilisation correcte de tempo.

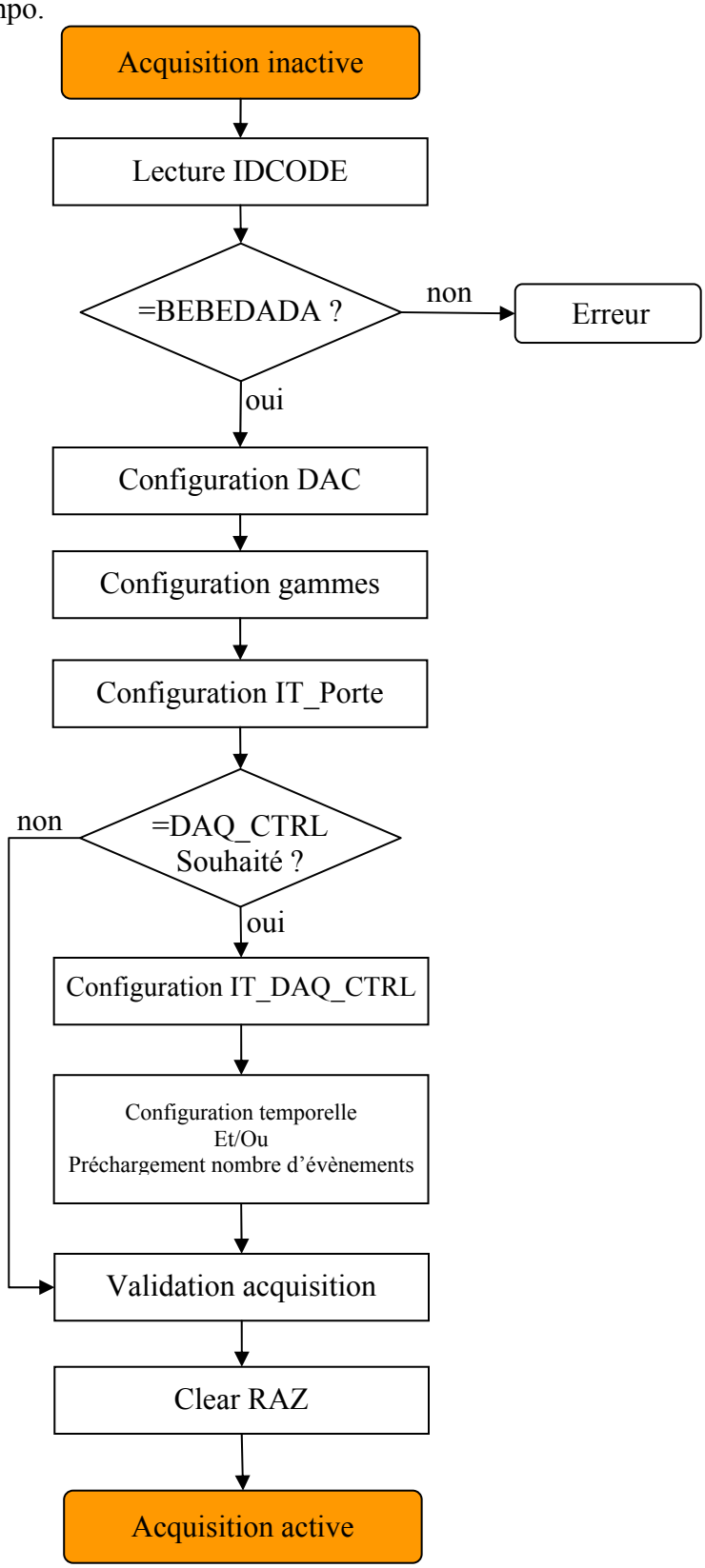


Figure 15 : Algorithme de programmation de tempo

L'algorithme suivant détaille l'utilisation de tempo lorsque l'acquisition est activée.

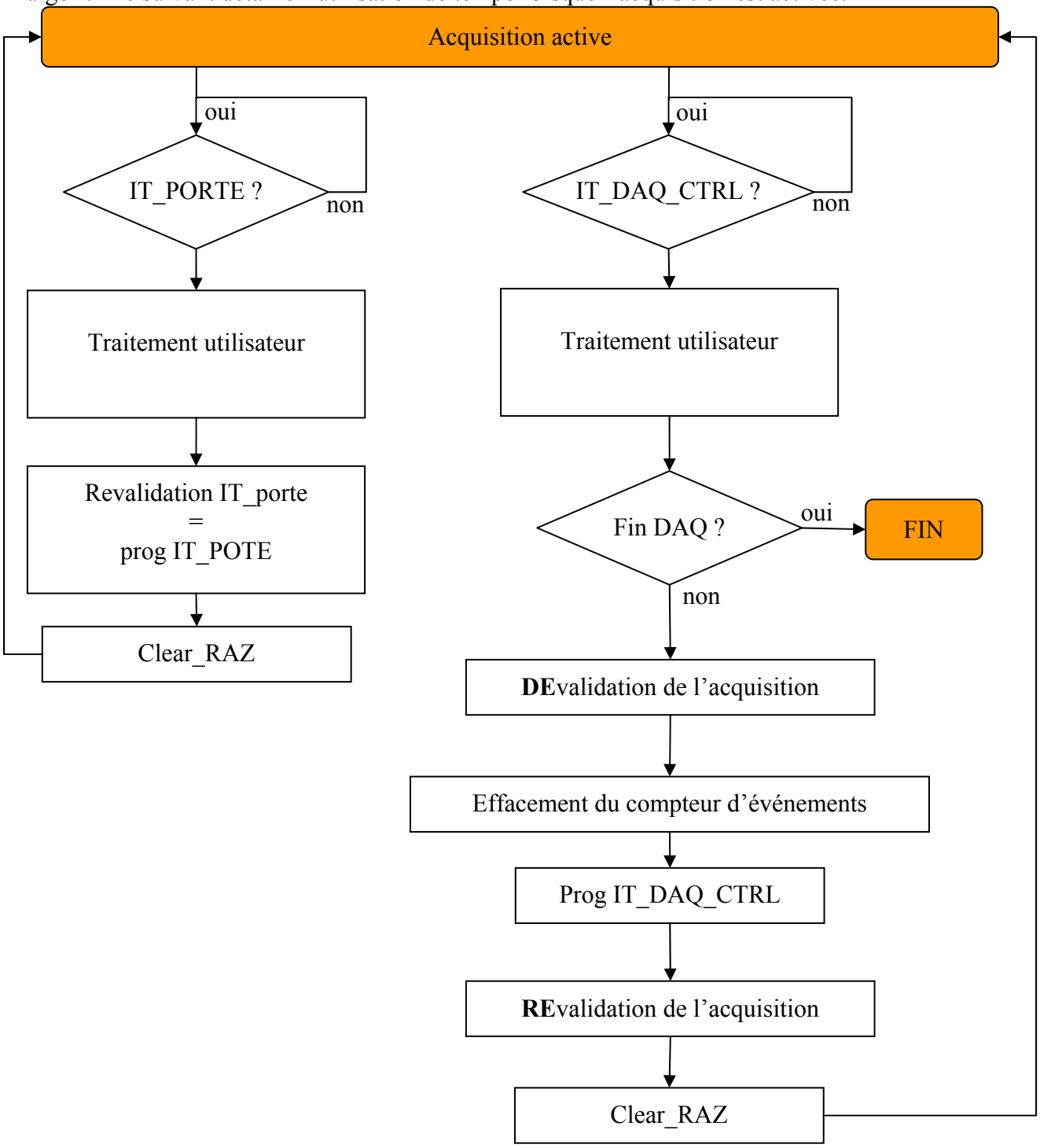


Figure 16 : algorithme du séquencement des acquisitions

3.8.3 Interface

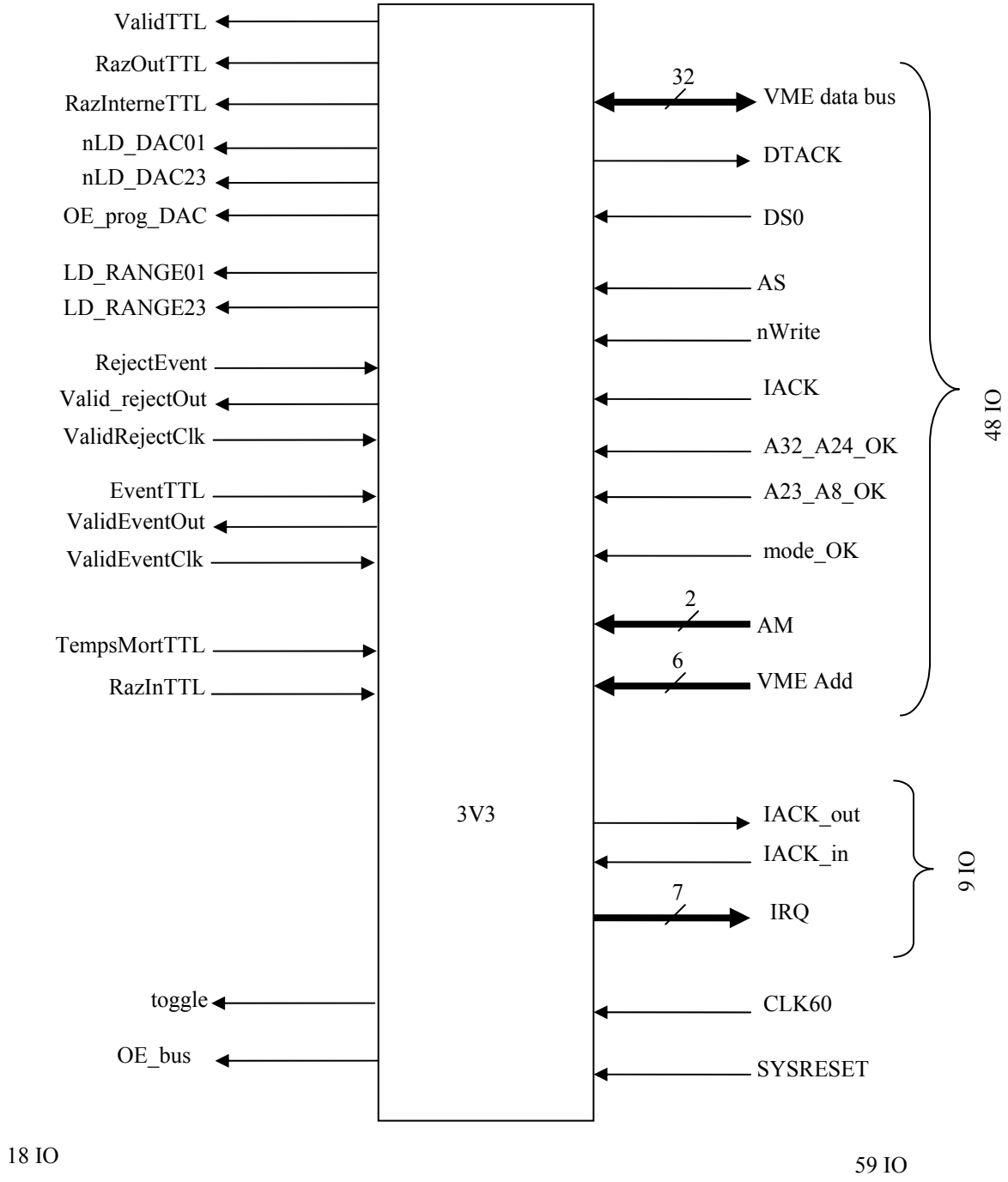


Figure 17 : Interface de l'EPLD TEMPO_VME

Total IO : 77

Mode_OK vérifie que AM[3..0] = $(1001)_{bin}$ et que LWORD='0' et que A[7]=$(0)_{bin}$ et DS1=$(0)_{bin}$

- **A32_A24_OK** : signale que la partie du bus d'adresse 32..24 est reconnue ;
- **A23_A8_OK OK** : signale que la partie du bus d'adresse 23..8 est reconnue ;
- **ValidTTL** : Permet la validation de la carte tempo sur une écriture VME ;
- **RazOutTTL** : Permet la génération du signal **RazOut** sur une écriture VME ;
- **RazInterneTTL** : Permet de reseter les 4 voies et d'assurer la décharge complète des condensateurs, il est déclenché sur une écriture VME ou sur la réception d'un signal **RazIn** ;
- **nLD_DAC01, nLD_DAC23** : Signaux permettant de charger les DAC avec les valeurs présentées sur le bus VME ;
- **OE_PROG** : utilisé en conjonction avec n**LD_DACxx** et **LD_RANGE**, valide le buffer transmettant le bus de données VME vers les DAC et les sélecteurs de calibre ;
- **LD_RANGE01, LD_RANGE23** : Signaux permettant le chargement des sélecteurs de calibre ;
- **Rejectevent** : signal TTL, indiquant qu'un événement a été rejeté ;
- **ValidRejectOut** : signal TTL, indiquant qu'un événement a été rejeté et que la carte est validée.
- **ValidRejectClk** : Signal d'entrée horloge, cablé en externe avec ValidRejectOut ;
- **EventTTL** : signal TTL indiquant qu'un évènement a eu lieu ;
- **ValidEventOut** : signal TTL indiquant qu'un évènement a eu lieu et que la carte est validée ;
- **ValidEventClk** : signal TTL, indiquant qu'un événement a eu lieu et que la carte est validée.
- **TempsMortTTL** : signal TTL, image du signal **TempsMort** Chaque passage de ce signal à '1' reflète la réception d'un évènement valide. Permet l'activation de l'interruption ;
- **RazInTTL** : signal TTL, indiquant l'activation en entrée du signal **RazIn** ;
- **OE_BUS** : validation du bus VME principal ;

3.8.4 Synoptique

La création d'un chip select interne appelé CS_int simplifie la lecture. Son équation est :

CS_int=(A32_A24_OK xnor AM[5..0]= « 00 ») and A24_A8_OK and Mode_OK and IACK and DS= « 00 »

Aussi CS_int passe par 2 bascules pour être resynchronisé, du coup cela laisse 2*60 ns pour le décodage et pour la stabilisation des signaux.

3.8.4.1 Module numéro de carte

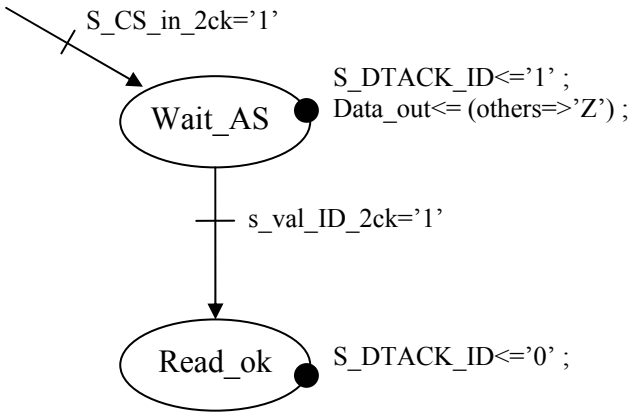


Figure 18 : FSM IDCODE

```
s_val_ID<='1' when (VME_ADD(6 downto 1)="000000" and nwrite='1') else '0';
```

3.8.4.2 Module mise en route acquisition

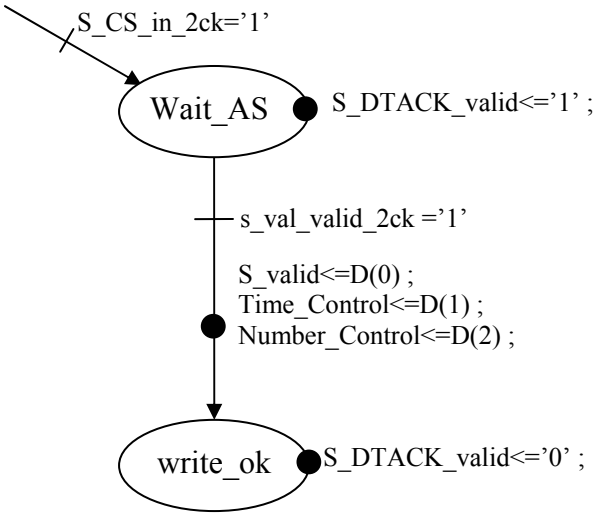


Figure 19 : FSM de validation

```
s_val_valid<='1' when (VME_ADD(6 downto 1)="000010" and nwrite='0') else
'0';
```

La gestion du contrôle d'acquisition est gérée par la machine suivante :

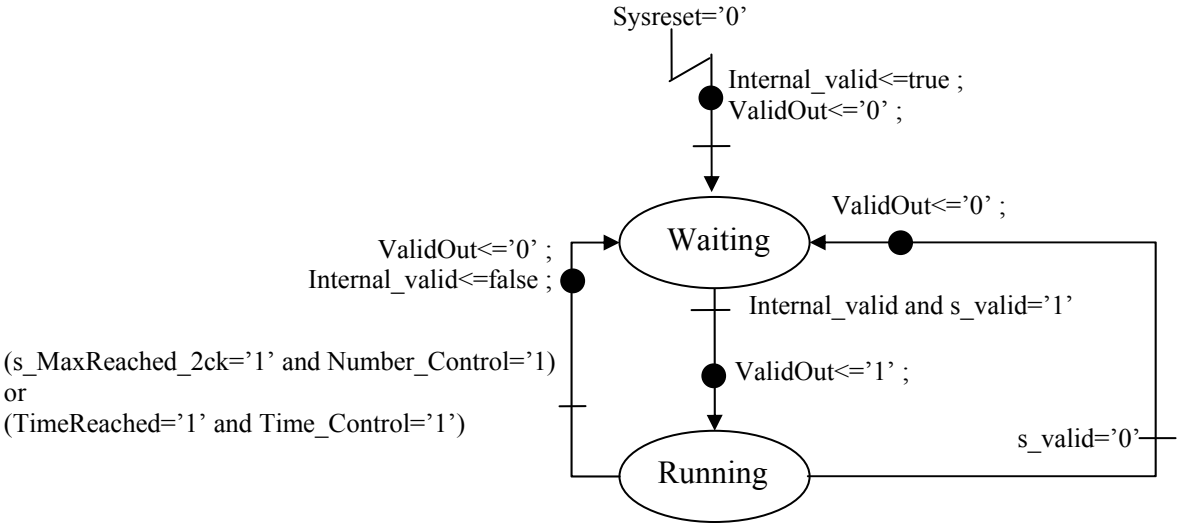


Figure 20 : FSM de gestion du contrôle d'acquisition

Internal_valid, sert à éviter le rebouclage incontrôlé.
Il est à noter que le mécanisme de revalidation de **Internal_valid** n'est pas explicité ici. En fait, lorsque un accès un VME correct est détecté par la FSM de validation, elle va passer par l'état **end_write**, et c'est ce passage qui revalide **Internal_valid**.

3.8.4.3 Module de configuration DAC

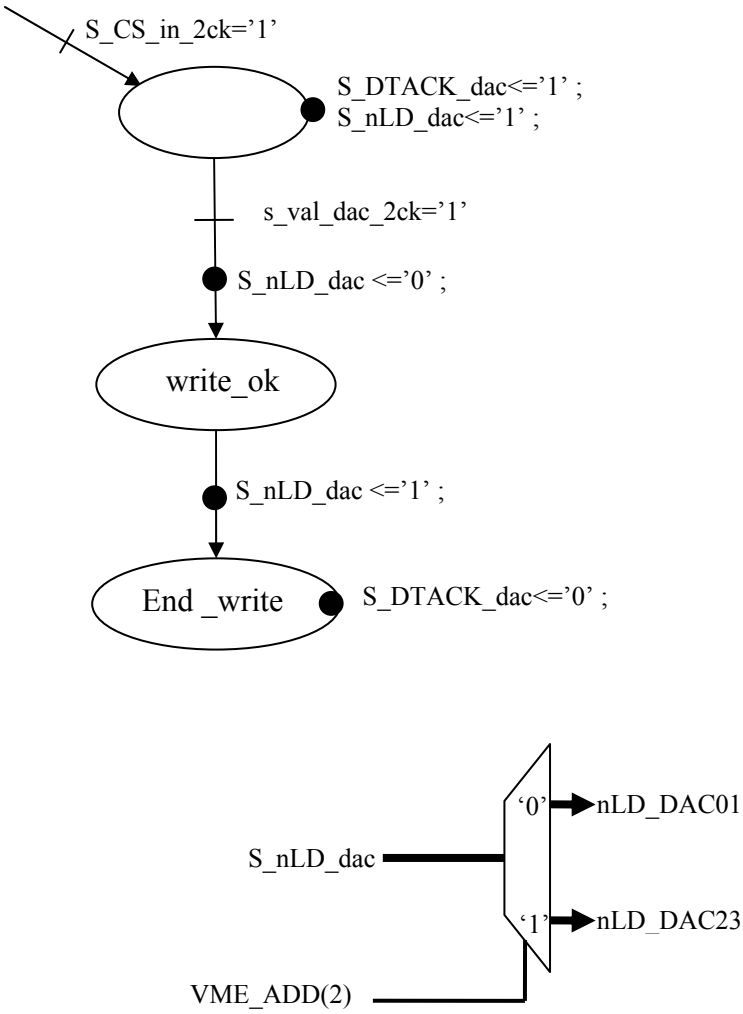


Figure 21 : FSM de programmation DAC

```
s_val_dac<='1' when (VME_ADD(6 downto 3)="0001" and nwrite='0' and
VME_ADD(1)='0') else '0';
```

Ensuite le signal **s_LD_dac** est aiguillé en fonction de **VME_ADD**(2).
Comme n**LD_dac** dure au moins 1 période, le temps minimum spécifié pour les DAC est respecté.

3.8.4.4 Module de RANGE_CTRL

Ce module sert à générer les signaux nécessaires pour charger les données dans des registres de type 74LVC374.

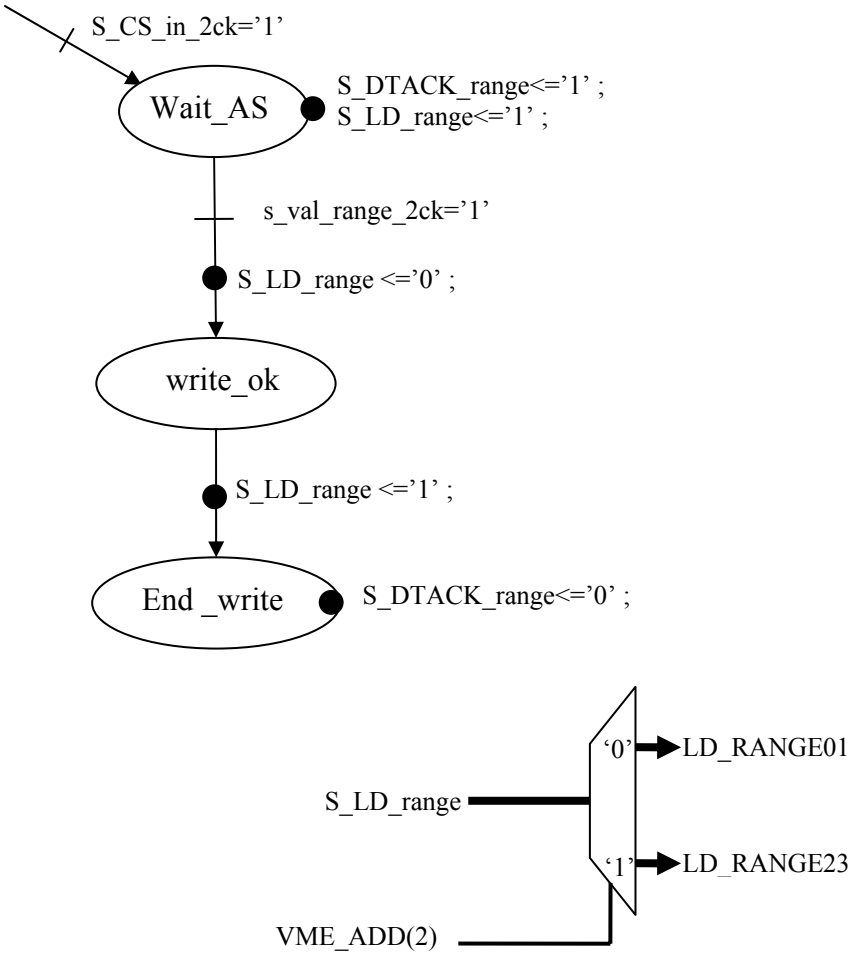


Figure 22 : FSM de programmation du calibre

```
s_val_range<='1' when (VME_ADD(6 downto 3)="0010" and nwrite='0' and
VME_ADD(1)='0') else '0';
```

Le signal s_LD_range est aiguillé en fonction de VME_ADD(2).

3.8.4.5 Module de clear RAZ

Ce module est composé de 3 parties :

- La partie sensible à l'écriture VME;
- La partie génératrice du RazOut
- La partie sensible au RazIn ;
- La partie génératrice de RefusRazIn ;
- La partie détectant la retombée des portes ;

Pour la compréhension, voir Figure 11 : Chronogramme de gestion du RazInterne, page 16 et Figure 12 : Chronogramme de gestion du RazInterne dans le cas de ClearRaz, page 17.

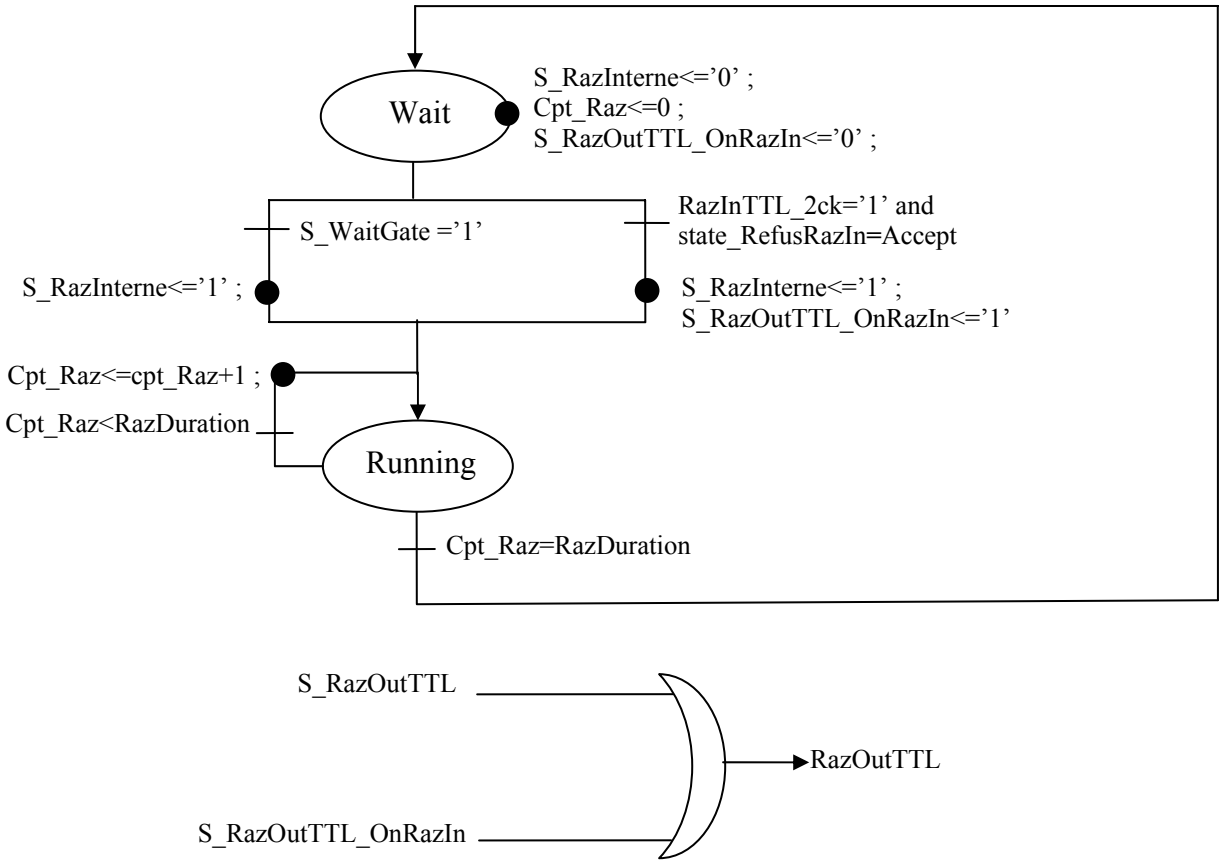


Figure 23 : FSM de gestion du RAZ interne

Le signal RazInTTL est resynchronisé par 2 bascules. Il faut noter que le RazInterne généré est le résultat d’un OU logique entre le signal s_RazInterne et le complément du signal de reset global.

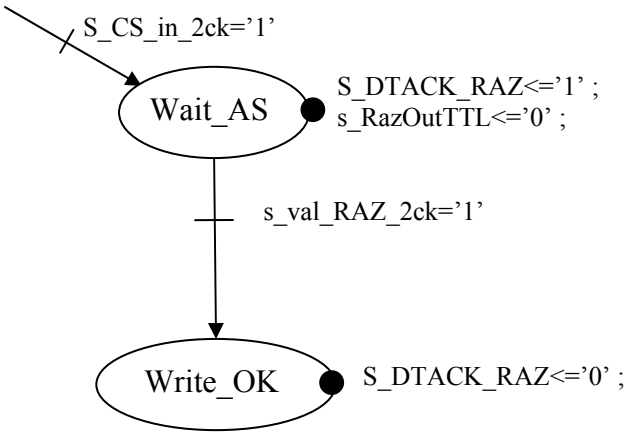


Figure 24 : FSM de gestion du VME

```
s_val_raz<='1' when (VME_ADD(6 downto 1)="001100" and nwrite='0') else '0';
```

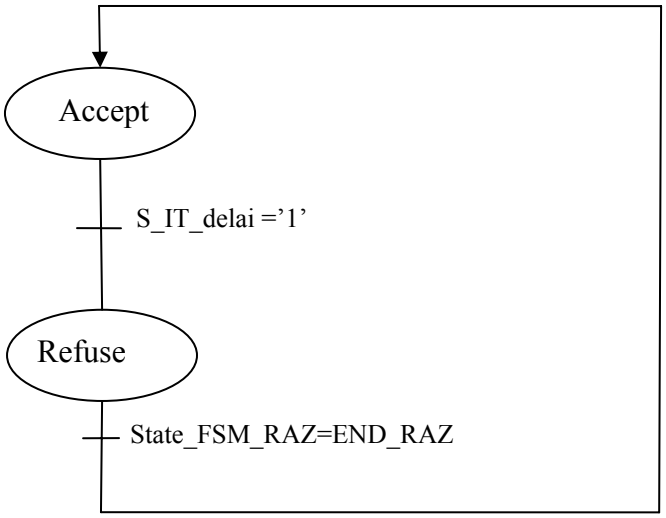


Figure 25 : FSM de gestion du RefusRazIn

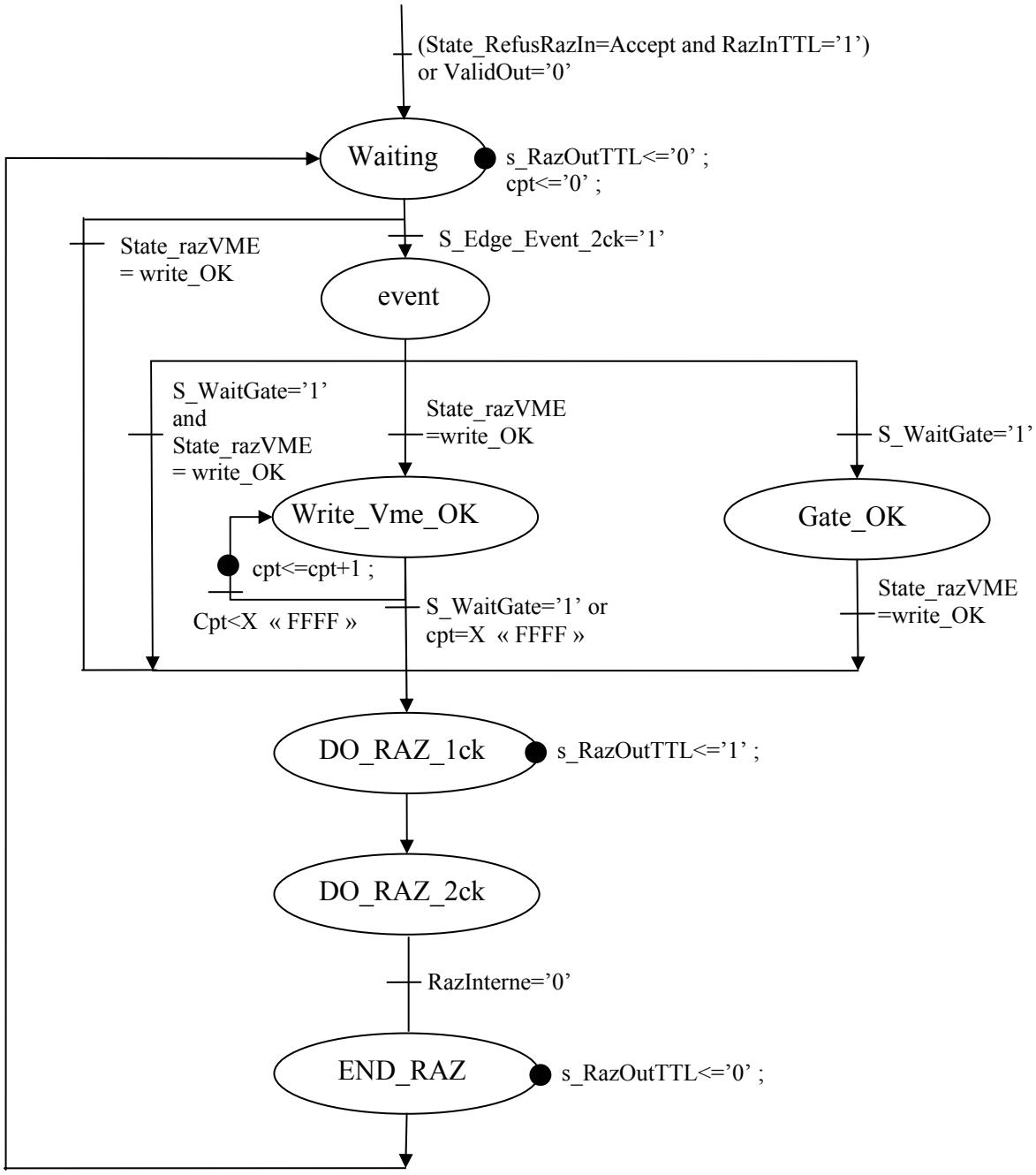


Figure 26 : FSM de gestion du RazOut

Cette FSM fait la différence entre le premier clear RAZ pour sortir du temps mort à l'init ou après une devalidation. Si un événement est détecté, suivant que l'ecriture VME arrive avant ou après la fin des portes, le raz out est généré.
Attention, si **s_waitGate** n'est pas généré, la carte est bloquée, cela peut être gênant lors de la recherche des minimums de largeur et de retard, car alors de temps en temps, ils ne sont pas générés et la carte reste bloquée en temps mort. Pour éviter les problèmes, dans le cas bloquant **write_vme_OK**, on prévoit le forçage de la sortie au bout de 4,096 ms … C'est une sorte de watchdog.

3.8.4.6 Module de programmation temporelle

Ce module comporte 2 parties :

- la partie d'écriture VME, qui permet de sélectionner le temps ;
- La partie compteur + comparateur ;

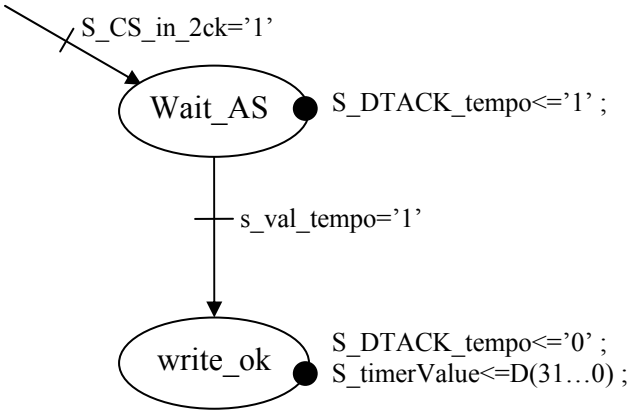


Figure 27 : FSM de programmation temporelle

```
s_val_tempo<='1' when (VME_ADD(6 downto 1)="001110" and nwrite='0') else '0';
```

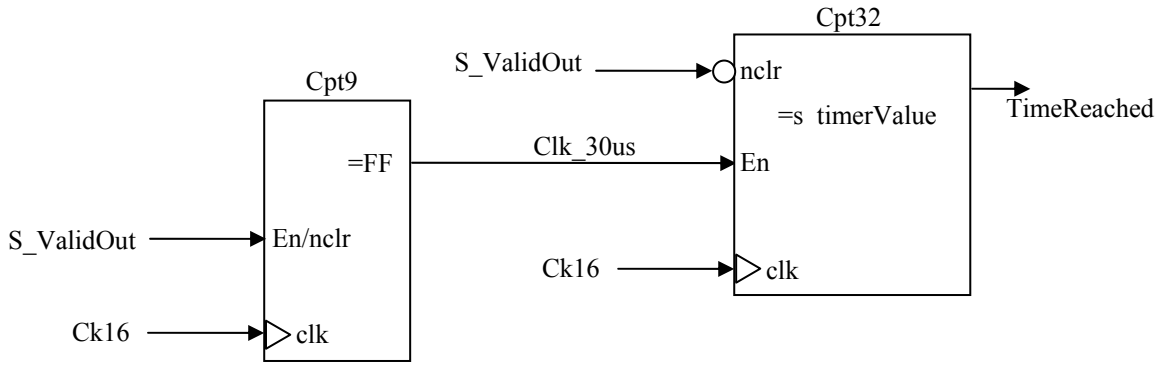


Figure 28 : Mécanisme de comptage temporel

Le mécanisme est mis en route uniquement lorsque le système est validé. Dès que la carte tempo est dévalidée, les 2 compteurs sont réinitialisés.

3.8.4.7 Module de lecture du nombre d'évènements rejetés

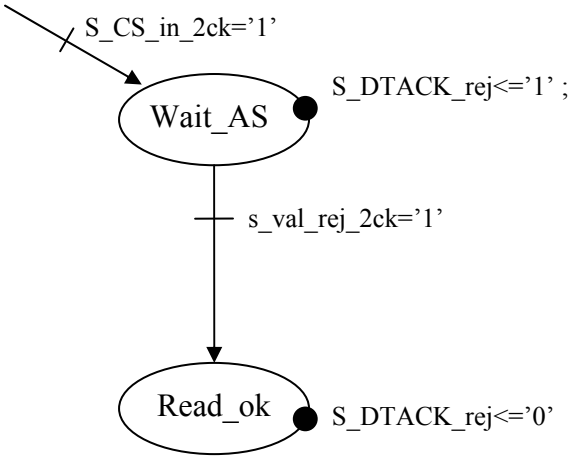


Figure 29 : FSM de lecture du nombre d'évènements rejetés

```
s_val_rej<='1' when (VME_ADD(6 downto 1)="010000" and nwrite='1') else '0';
```

De plus, le signal **OE_RejectCounter** permettant de passer le compteur d'événements rejetés en basse impédance est généré de façon asynchrone.

```
OE_RejectCounter<='1' when (AS='0' and s_CS_int='1' and s_val_rej='1') else '1';
```

Partie compteur

C'est un compteur synchrone effaçable, incrémenté sur chaque front montant du signal RejectEventTTL. Il fonctionne à au moins 50 MHz (soit le pire cas 20ns de largeur d'impulsion suivit d'une autre). L'incrément se passera bien si entre 2 événements il se passe au moins 4 ns. Il est relu à travers le VME.

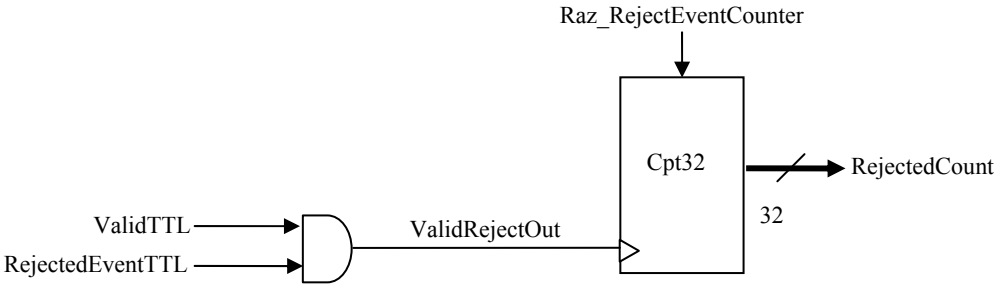


Figure 30 : compteur d'événements rejetés

3.8.4.8 Module de lecture du nombre d'événement

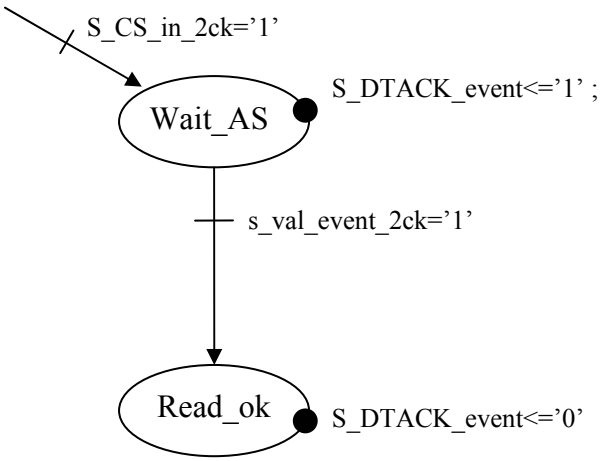


Figure 31 : FSM de lecture du nombre d'évènements

s_val_event<='1' **when** (VME_ADD(6 **downto** 1)="010010" **and** nwrite='1') **else** '0';
De la même façon que précédemment le signal **OE_EventCounter** est généré de façon asynchrone.

```
OE_EventCounter<='1' when (AS='0' and s_CS_int='1' and s_val_event='1') else '1';
```

Partie compteur et comparateur

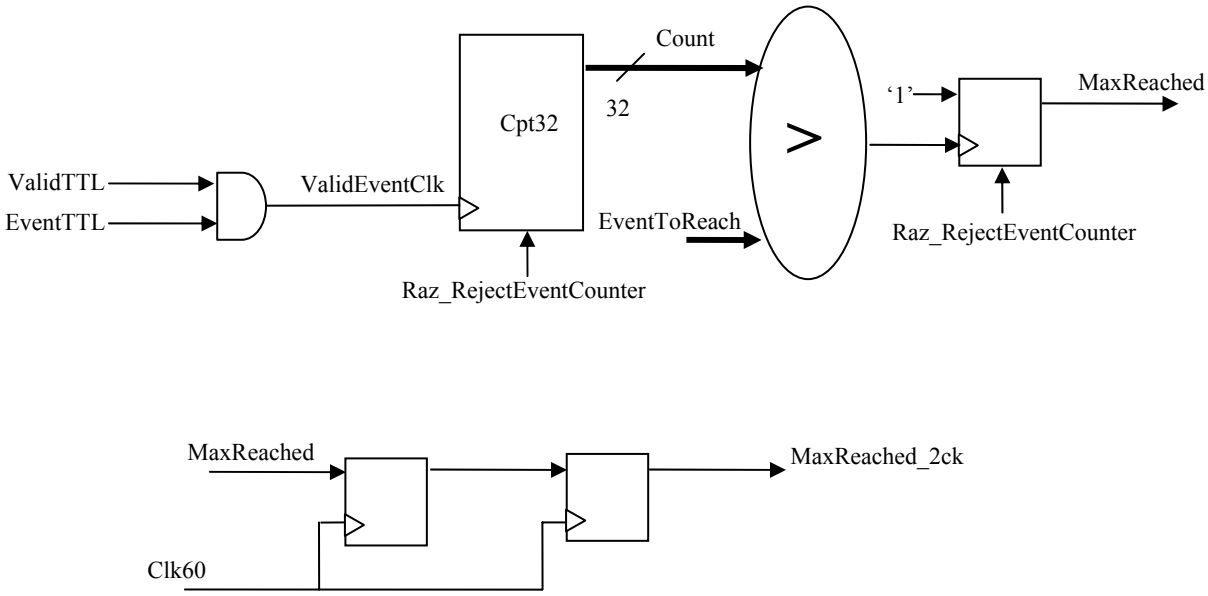


Figure 32 : compteur d'événement avec contrôle d'acquisition

3.8.4.9 Module de préchargement/reset du nombre d'évènement

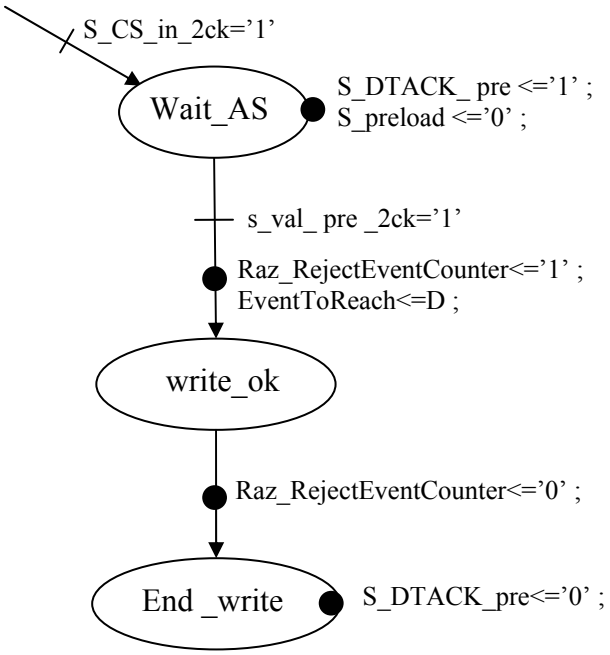


Figure 33 : FSM de préchargement/reset

```
s_val_pre<='1' when (VME_ADD(6 downto 1)="010100" and nwrite='0') else '0';
```

3.8.4.10 Module de programmation et de gestion des interruptions

3.8.4.10.1 Validation de l'IT porte

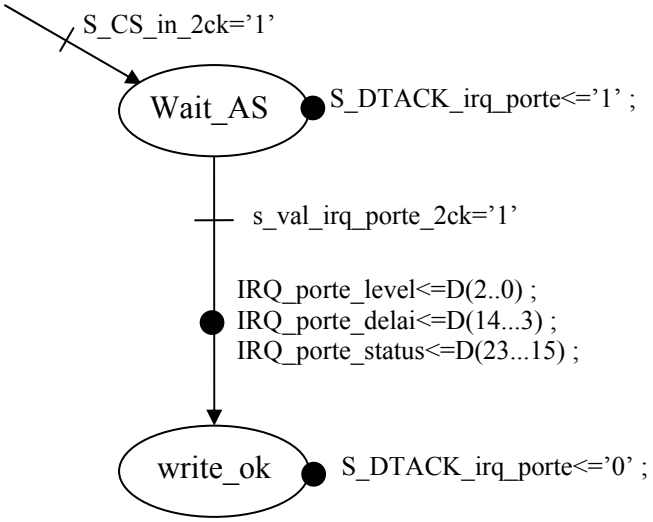


Figure 34 : FSM de validation d'interruption porte

```
s_val_irq_porte<='1' when (VME_ADD(6 downto 1)="010110" and nwrite='0') else '0';
```

3.8.4.10.2 Validation de l'IT DAQ_CTRL

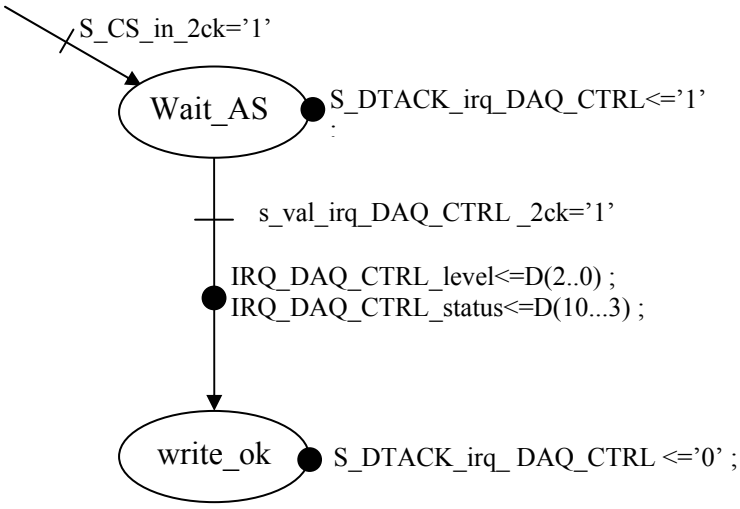


Figure 35 : FSM de validation d'interruption DAQ_CTRL

s_val_irq_DAQ_CTRL<='1' **when** (VME_ADD(6 **downto** 1)="011000" **and** nwrite='0') **else** '0';

3.8.4.10.3 Chaînage des IT

Un retour est volontairement ajouté sur IACKIN pour avoir le temps de faire le traitement avant de propager vers IACKOUT

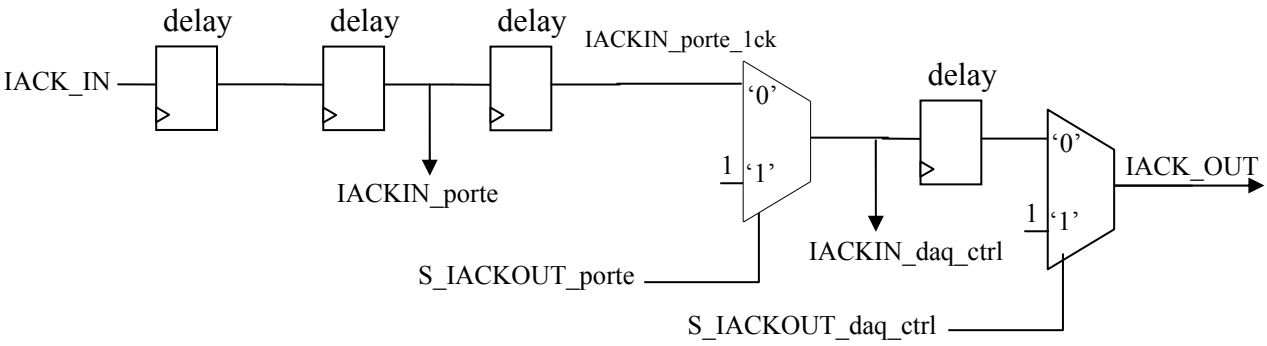


Figure 36 : gestion du chaînage des **IT**

Ainsi si la carte ne répond pas à l'interruption, elle transmet le '0', sinon elle transmet '1' et répond à l'IT.

3.8.4.10.4 Gestion des IT

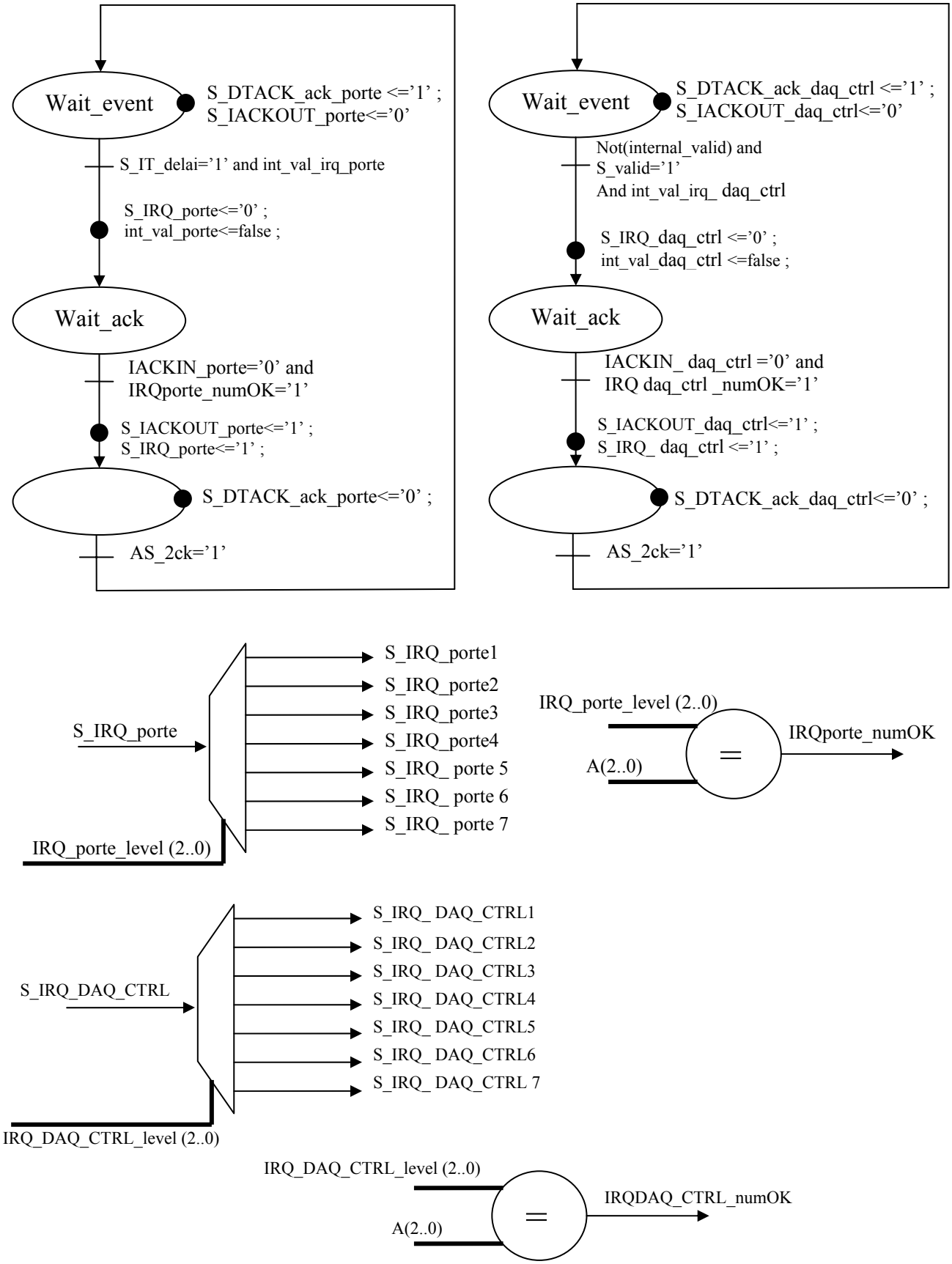


Figure 37 : gestion du cycle d'interruption

Lorsque la machine de gestion des interruptions détecte une interruption, elle invalide automatiquement l'interruption (en affectant int_val_XX à false), de cette manière le rebouclage en IT est évité. Donc après avoir traité l'IT incriminée, il faut réécrire dans le registre de programmation de l'IT concernée.

L'aiguillage en sortie de la donnée status est fait grâce à l'état courant des machines d'état (l'état **IRQ_ACK**). Suivant celle qui est dans cet état on sait quel status retourner.

3.8.4.10.5 Génération du délai entre l'évènement valide et l'IT porte

La réception d'un évènement valide est détectée grâce au **passage** à 1 du signal **TempsMort_TTL**. Ce signal reflète exactement la réception d'un évènement valide excepté à une seule occasion, la dévalidation de tempo, à savoir ValidTTL='0'. Il suffit d'empêcher la détection du front montant dans ce cas là.
Aussi, la génération d'IT doit s'arrêter si un **RazIn** valide est reçu.

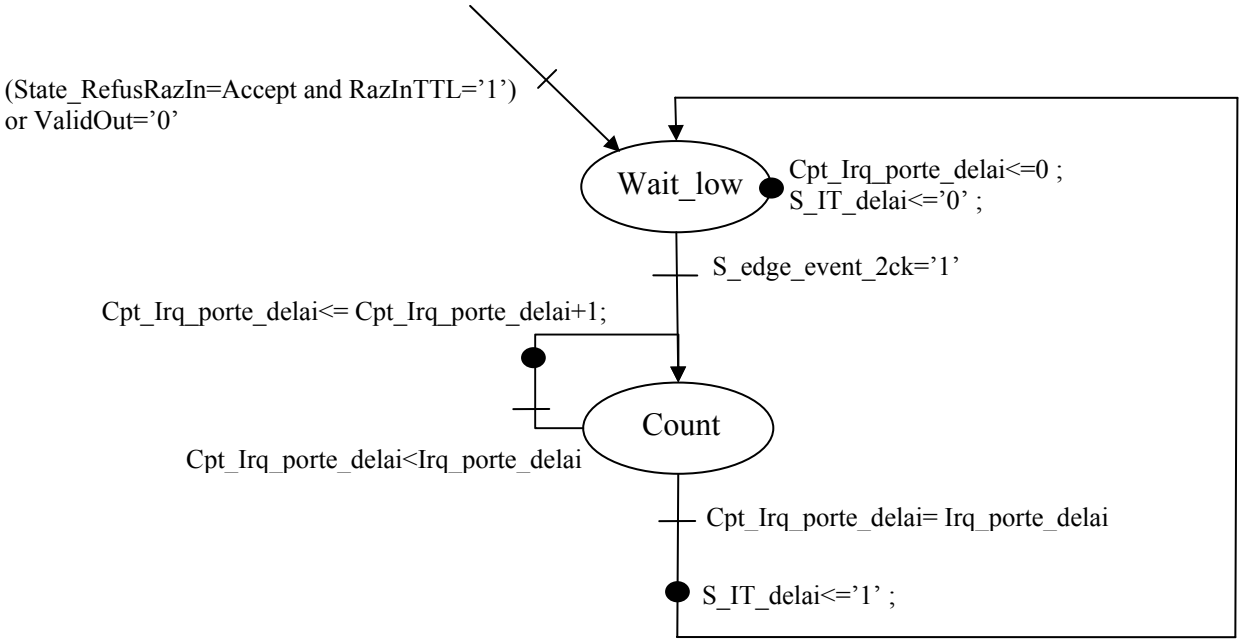
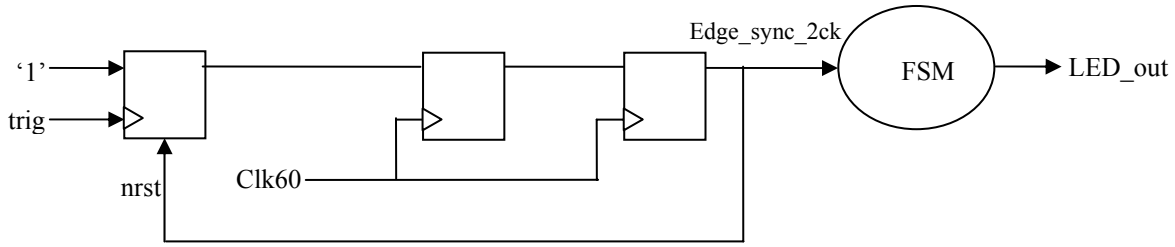


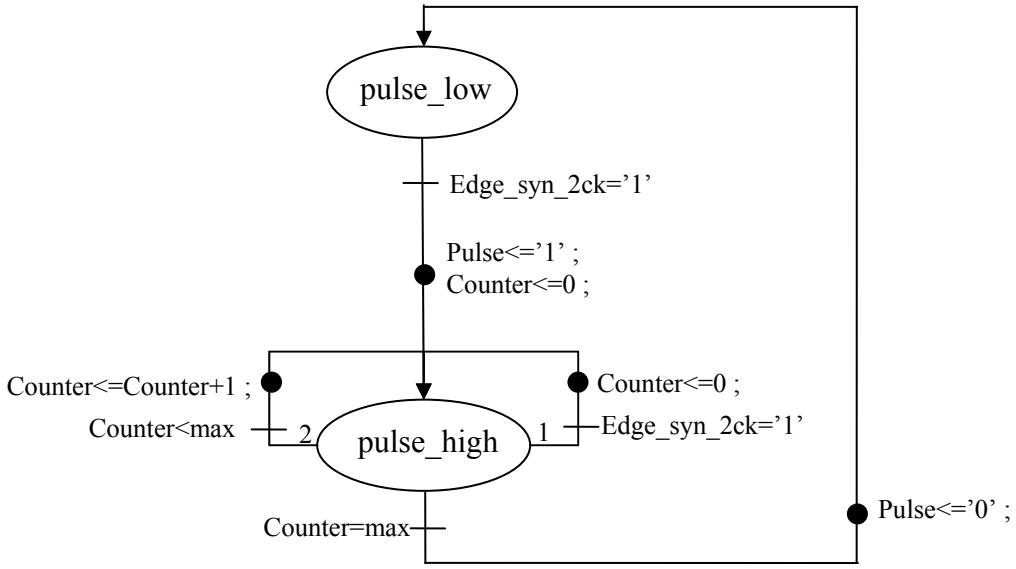
Figure 38 : FSM de génération de l'IT porte

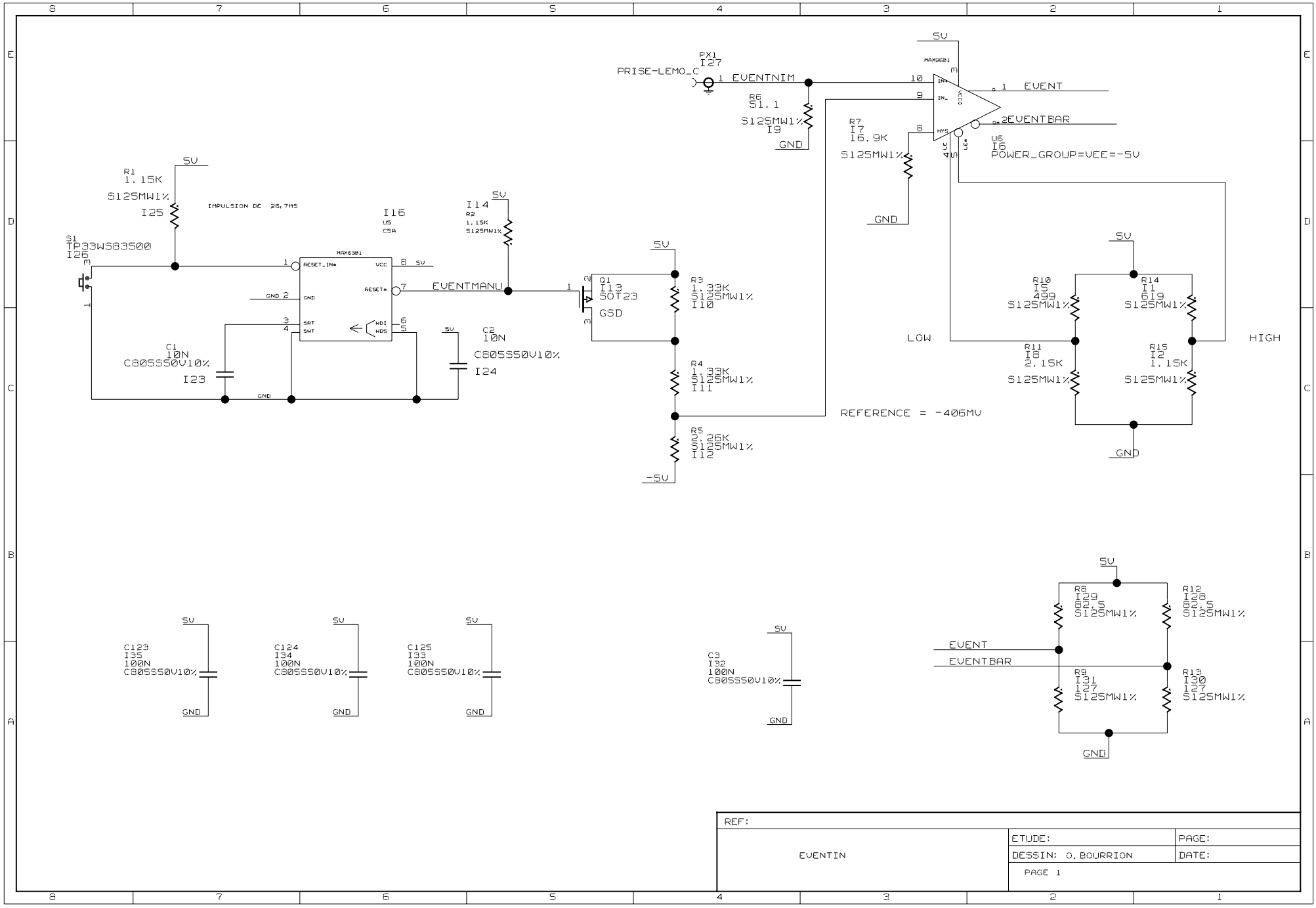
3.8.4.10.6 Monostables d'activité

Pour avoir une visualisation de l'arrivée des événements et de la mise en temps mort de la carte, il faut allumer des LED pendant un minimum de 100 ms pour pouvoir les voir … La détection d'évènement se fera sur front et un compteur redeclenchable est activé :

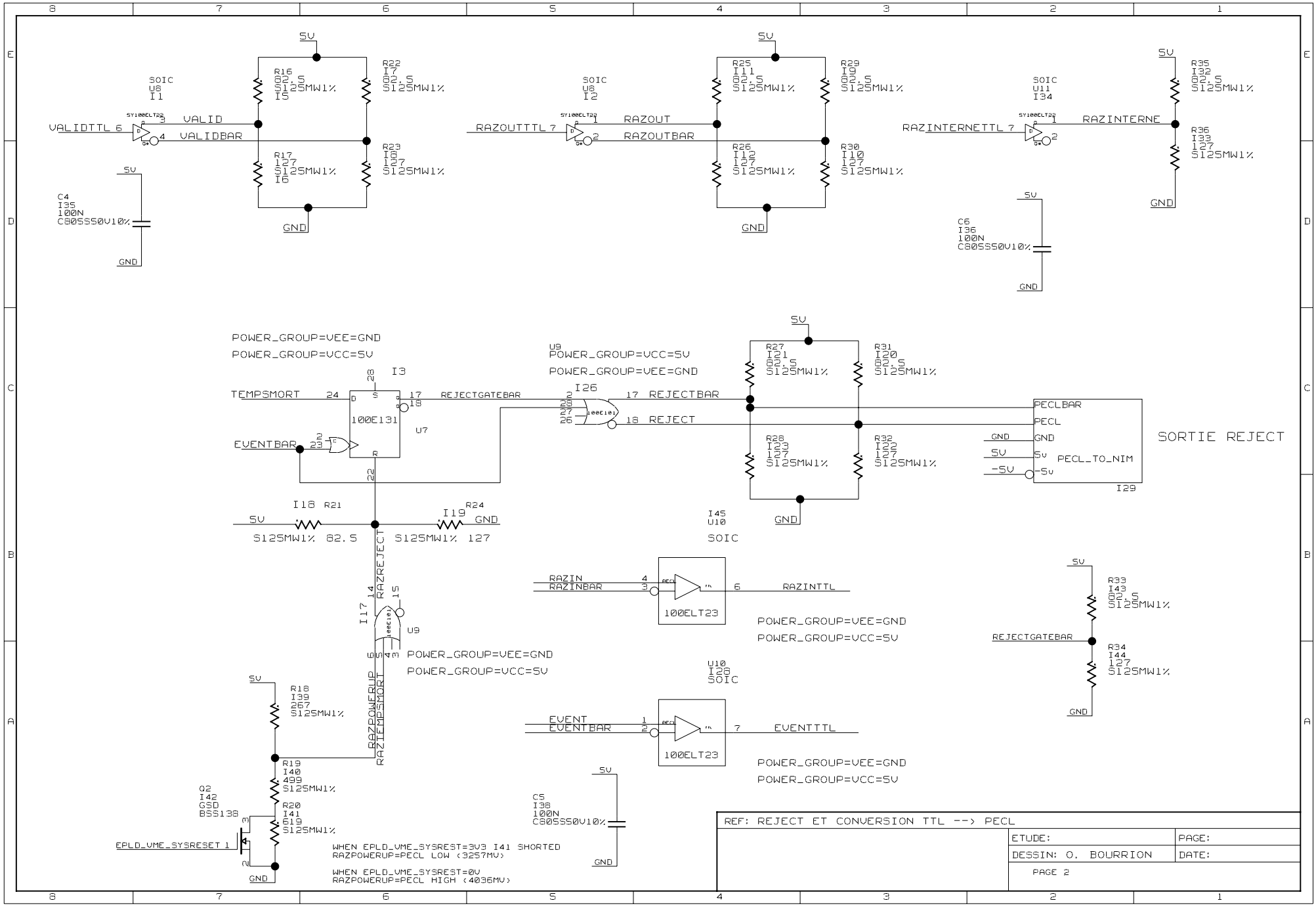
Le compteur doit compter sur 21 bits. 2^{21}*62,5ns=131ms

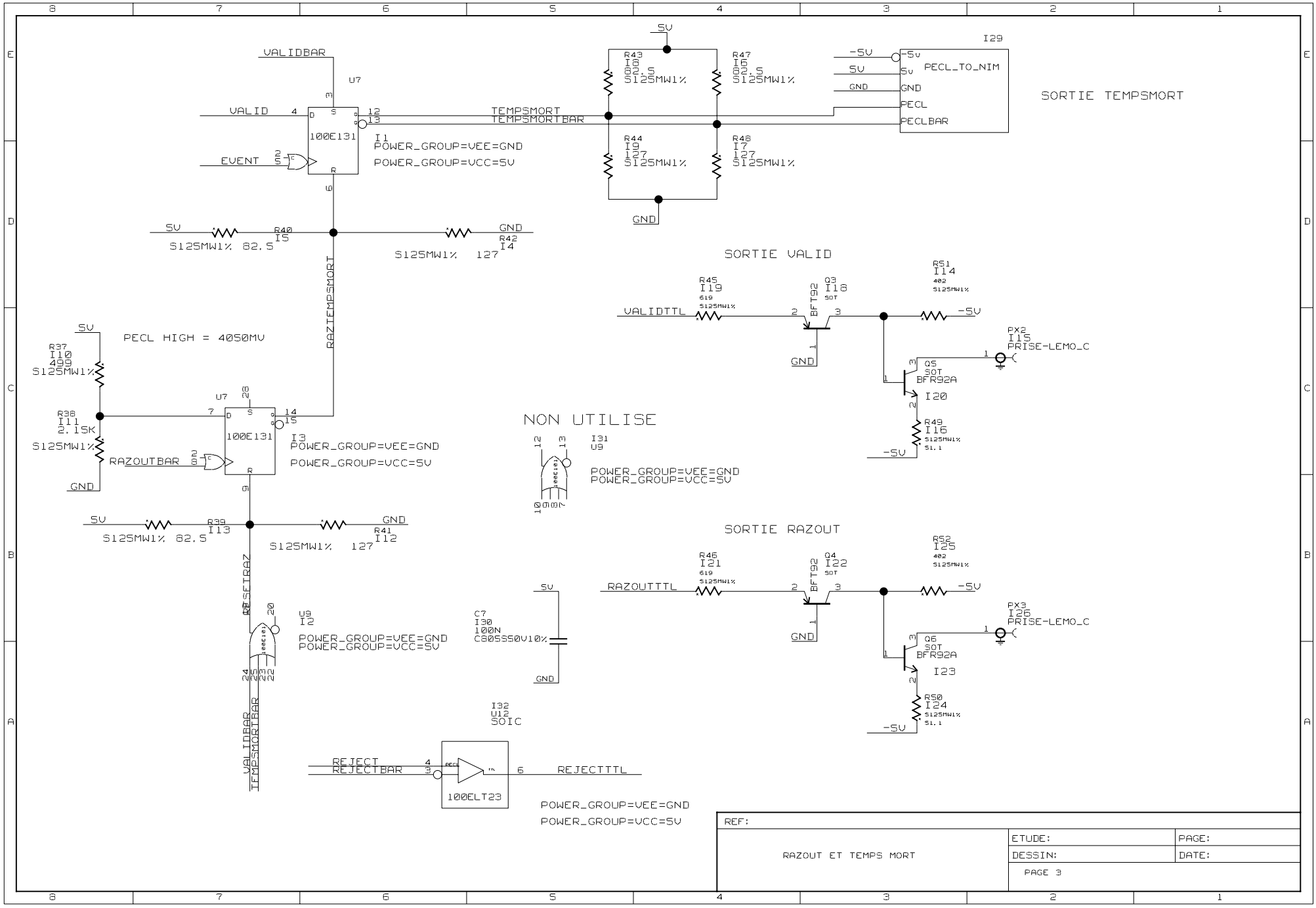

'1'
trig
nrst
Clk60
Edge_sync_2ck
FSM
LED_out


pulse_low
Edge_syn_2ck='1'
Pulse<='1' ;
Counter<=0 ;
Counter<=Counter+1 ;
Counter<max
2
pulse_high
1
Counter<=0 ;
Edge_syn_2ck='1'
Counter=max
Pulse<='0' ;

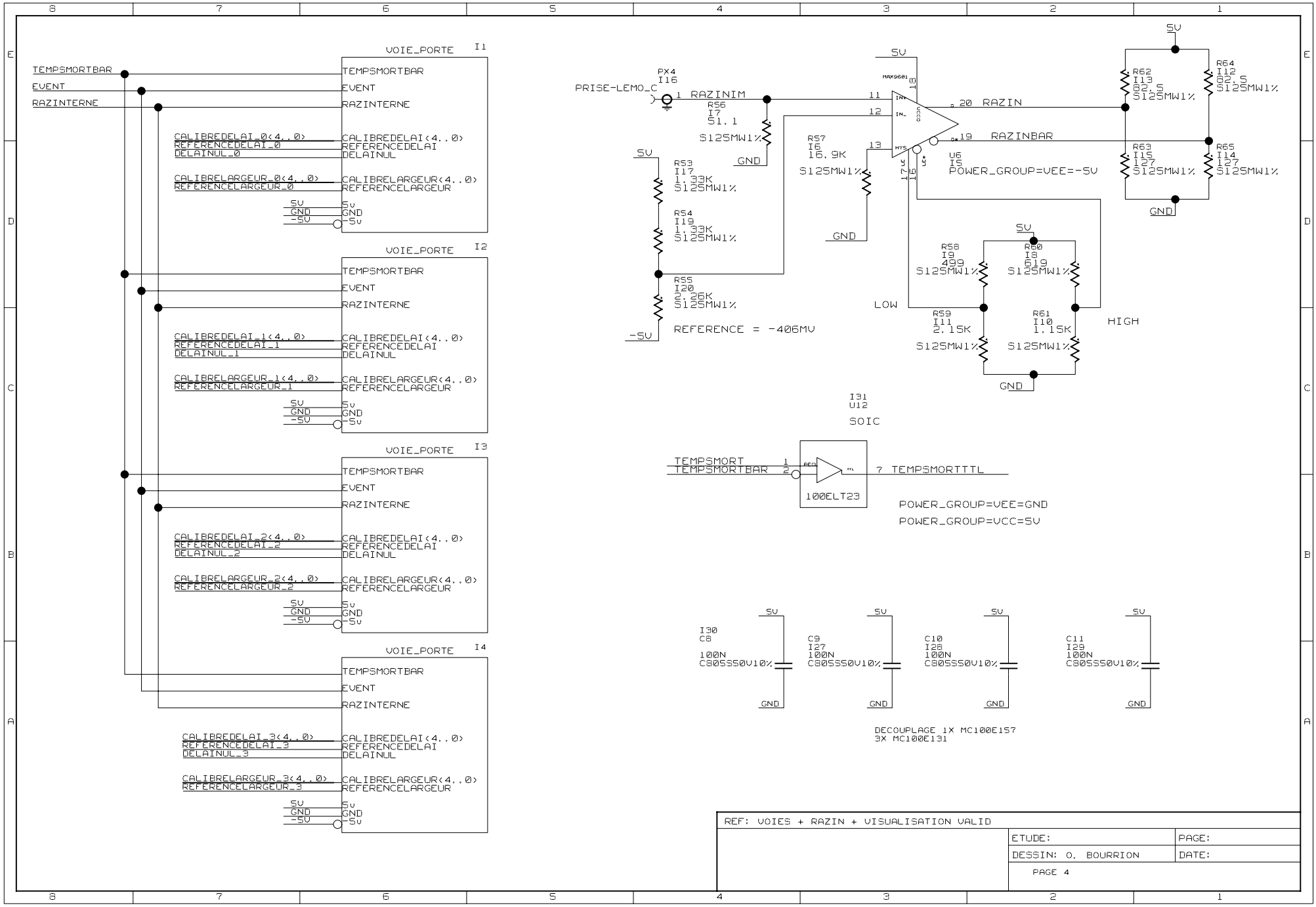


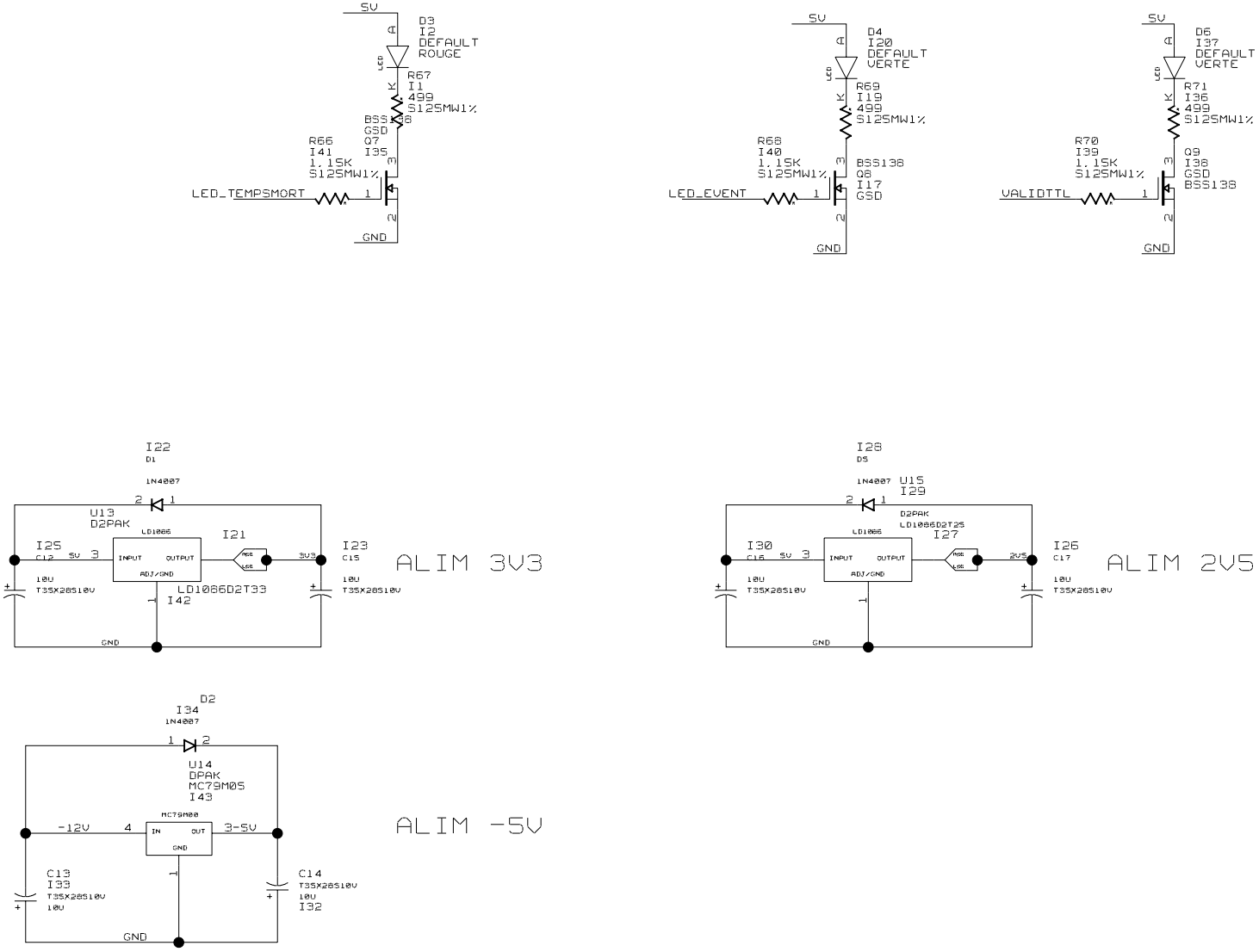
REF:		
EVENTIN	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:
	PAGE 1	


SOIC
U8
I1
VALIDTTL 6
VALID
VALIDBAR
R16
82.5
S125MW1%
I5
R17
127
S125MW1%
I6
R22
82.5
S125MW1%
R23
127
S125MW1%
C4
I35
100N
C805S50V10%
GND
SOIC
U8
I2
RAZOUTTTL 7
RAZOUT
RAZOUTBAR
R25
R26
R29
R30
SOIC
U11
I34
RAZINTERNETTTL 7
RAZINTERNE
R35
R36
C6
I36
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=SV
I3
TEMPSMORT
REJECTGATEBAR
100E131
U7
EVENTBAR
U9
POWER_GROUP=VCC=SV
POWER_GROUP=VEE=GND
I26
REJECTBAR
REJECT
R27
R28
R31
R32
PECLBAR
PECL
GND
SV
-SV
PECL_TO_NIM
I29
SORTIE REJECT
I18 R21
82.5
I19 R24
127
RAZREJECT
I17
RAZPOWERUP
RAZTEMPSMORT
I45
U10
SOIC
RAZIN
RAZINBAR
100ELT23
RAZINTTL
R33
I43
R34
I44
U10
I25
EVENT
EVENTBAR
EVENTTTL
R18
I39
257
R19
I40
499
R20
I41
619
Q2
I42
BSS138
EPLD_VME_SYSRESET 1
WHEN EPLD_VME_SYSREST=3V3 I41 SHORTED
RAZPOWERUP=PECL LOW (3257MV)
WHEN EPLD_VME_SYSREST=0V
RAZPOWERUP=PECL HIGH (4036MV)
C5
I38
100N
C805S50V10%
REF: REJECT ET CONVERSION TTL --> PECL
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:
PAGE 2

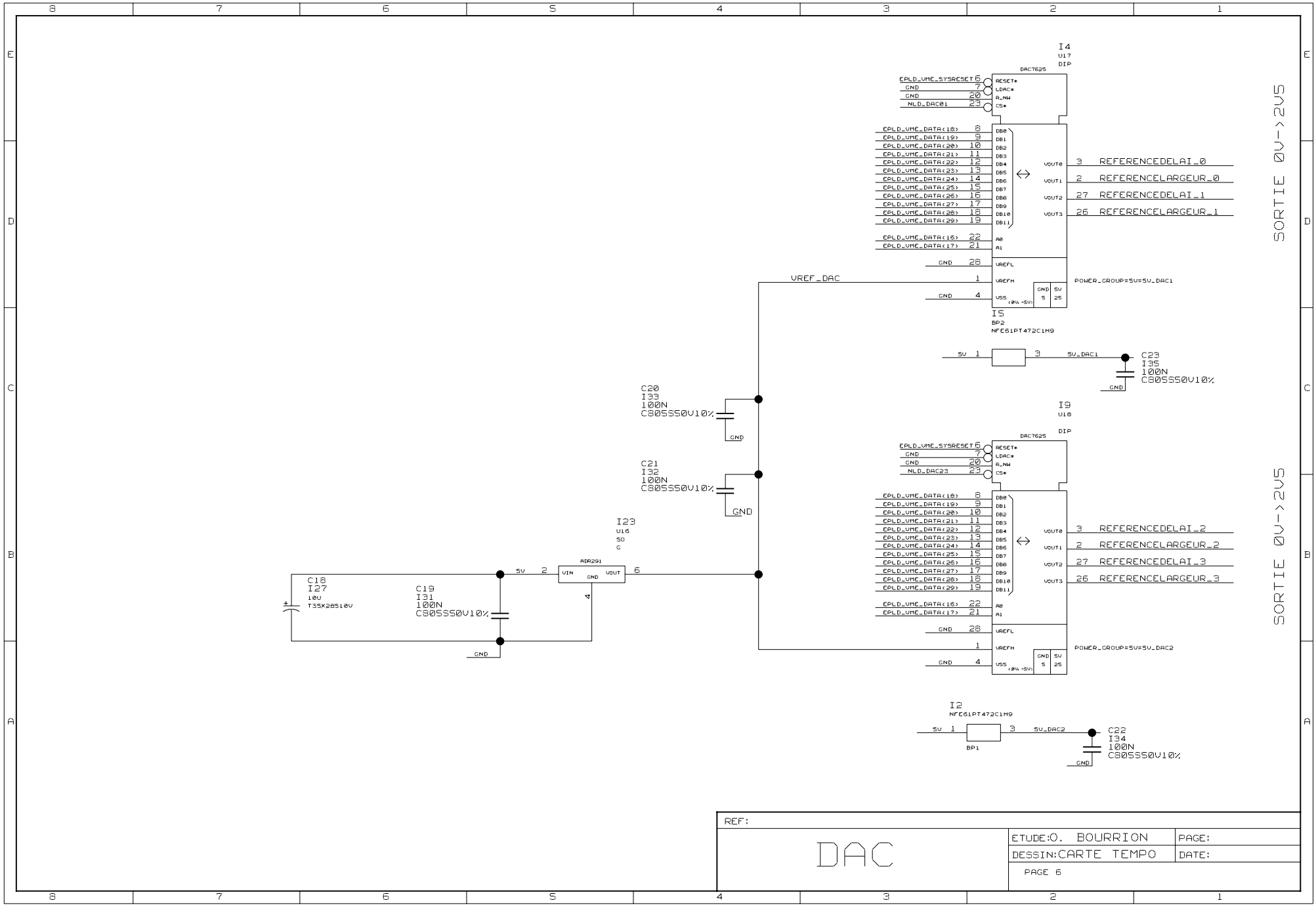


REF:		
RAZOUT ET TEMPS MORT	ETUDE:	PAGE:
	DESSIN:	DATE:
	PAGE 3	

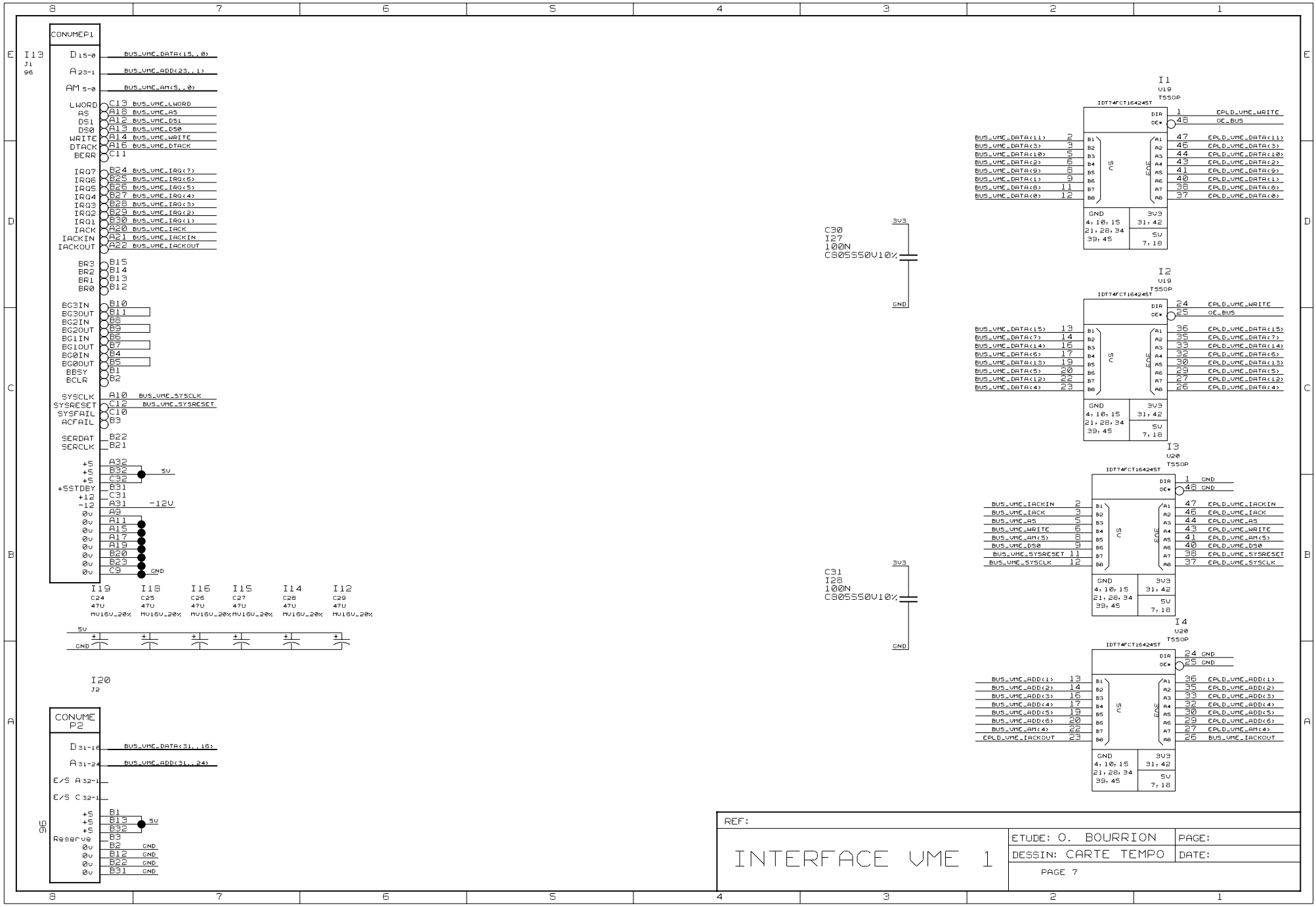

VOIE_PORTE I1
TEMPSMORTBAR
EVENT
RAZINTERNE
CALIBREDELAI_0<4..0>
REFERENCEDELAI_0
DELAINUL_0
CALIBRELARGEUR_0<4..0>
REFERENCELARGEUR_0
5V
GND
-5V
VOIE_PORTE I2
CALIBREDELAI_1<4..0>
REFERENCEDELAI_1
DELAINUL_1
CALIBRELARGEUR_1<4..0>
REFERENCELARGEUR_1
VOIE_PORTE I3
CALIBREDELAI_2<4..0>
REFERENCEDELAI_2
DELAINUL_2
CALIBRELARGEUR_2<4..0>
REFERENCELARGEUR_2
VOIE_PORTE I4
CALIBREDELAI_3<4..0>
REFERENCEDELAI_3
DELAINUL_3
CALIBRELARGEUR_3<4..0>
REFERENCELARGEUR_3
PX4
I16
PRISE-LEMO_C
RAZINIM
R56
I7
51.1
5125MW1%
R53
I17
1.33K
5125MW1%
R54
I19
1.33K
5125MW1%
R55
I20
2.26K
5125MW1%
REFERENCE = -406MV
MAX9601
RAZIN
RAZINBAR
R57
I6
16.9K
5125MW1%
U6
I5
POWER_GROUP=VEE=-5V
R62
I13
82.5
5125MW1%
R64
I12
82.5
5125MW1%
R63
I15
127
5125MW1%
R65
I14
127
5125MW1%
R58
I9
499
5125MW1%
R60
I8
619
5125MW1%
R59
I11
2.15K
5125MW1%
R61
I10
1.15K
5125MW1%
LOW
HIGH
I31
U12
SOIC
TEMPSMORT
TEMPSMORTBAR
100ELT23
TEMPSMORTTTL
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
I30
C8
100N
C805S50V10%
C9
I27
100N
C805S50V10%
C10
I28
100N
C805S50V10%
C11
I29
100N
C805S50V10%
DECOUPLAGE 1X MC100E157
3X MC100E131
REF: VOIES + RAZIN + VISUALISATION VALID
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:
PAGE 4

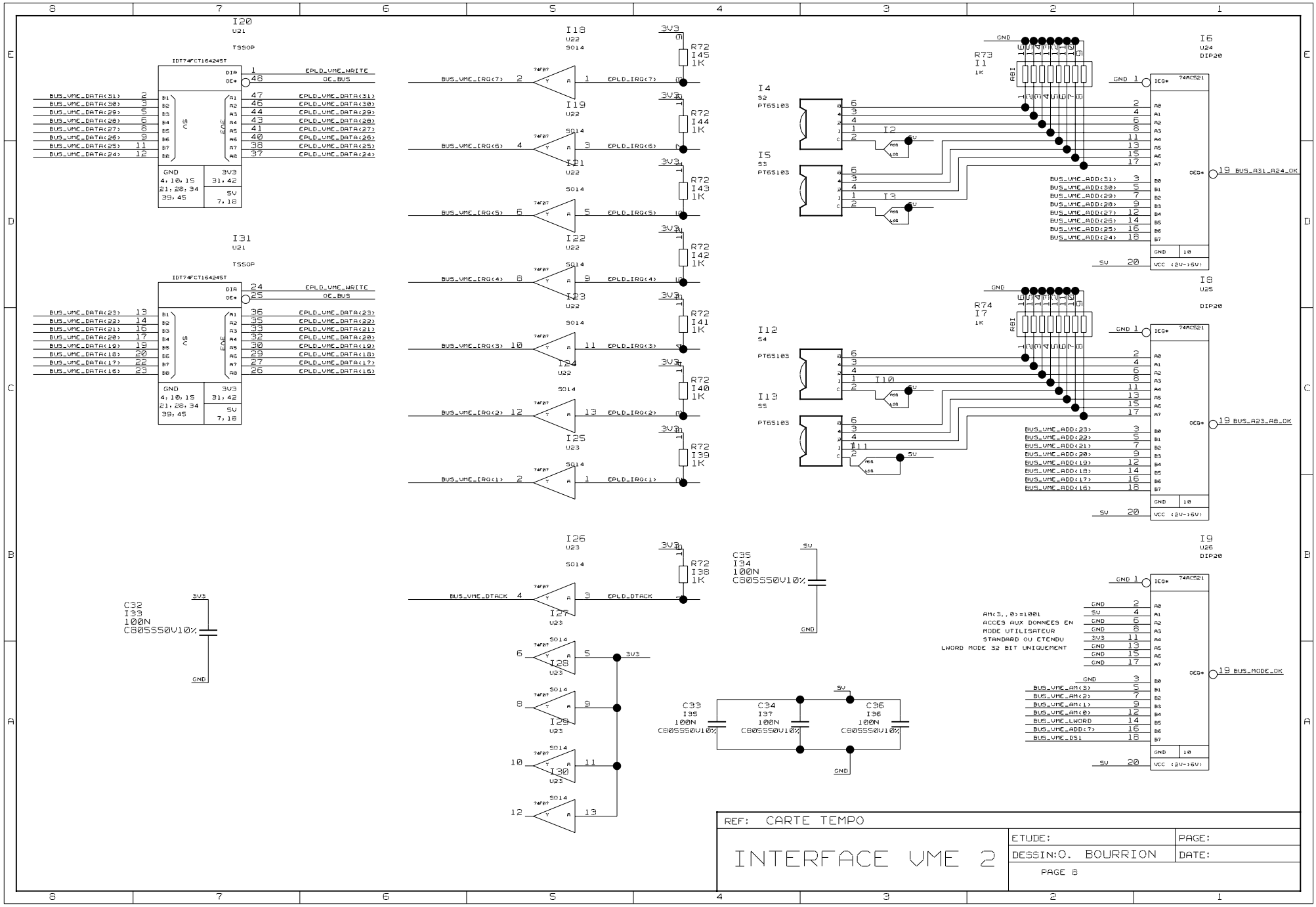


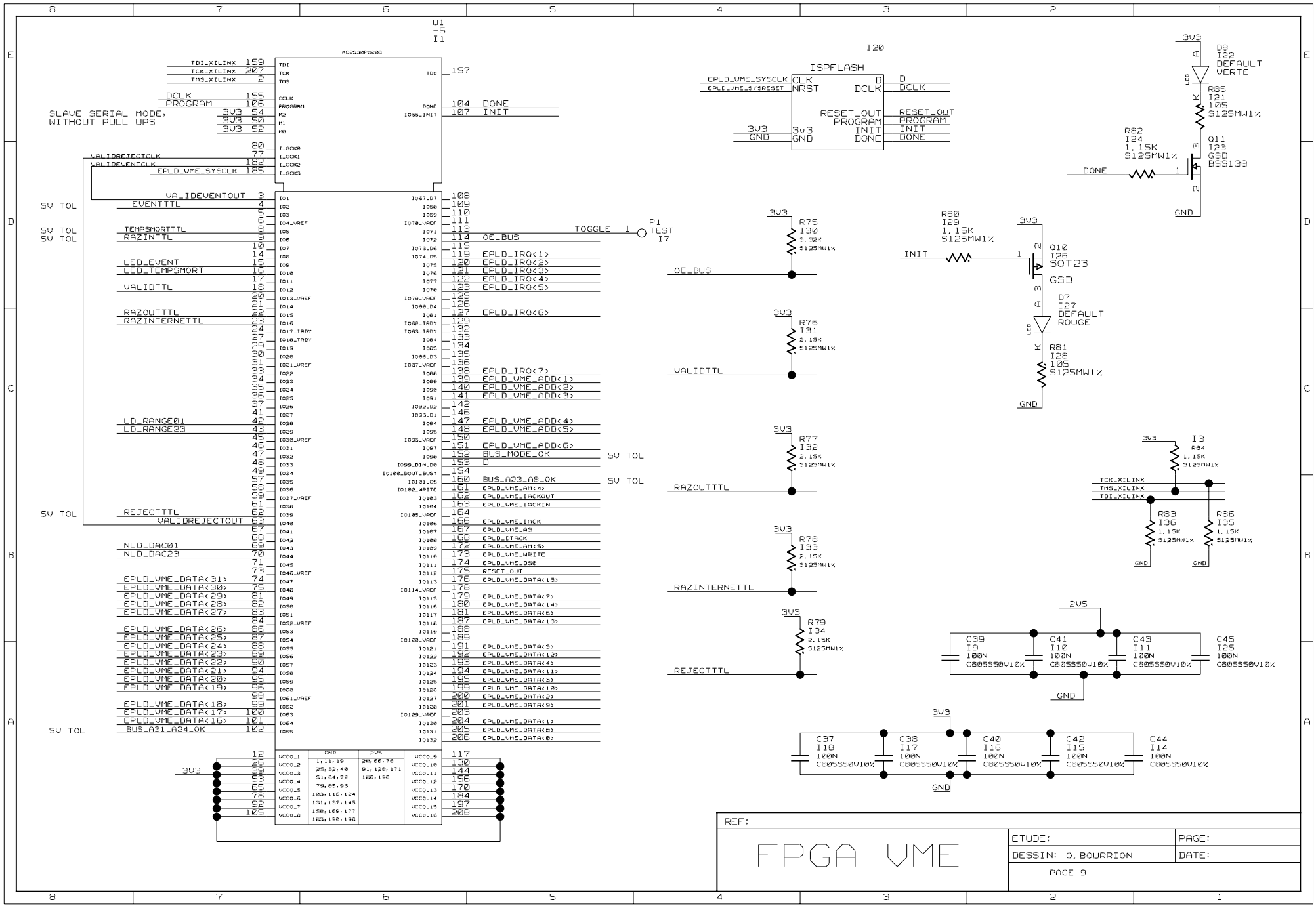
REF: VISUALISATION TEMPS MORT ET EVENT + ALIMENTATIONS		
	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:
	PAGE 5	

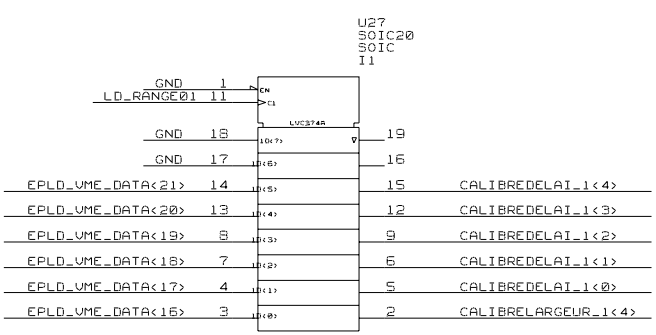


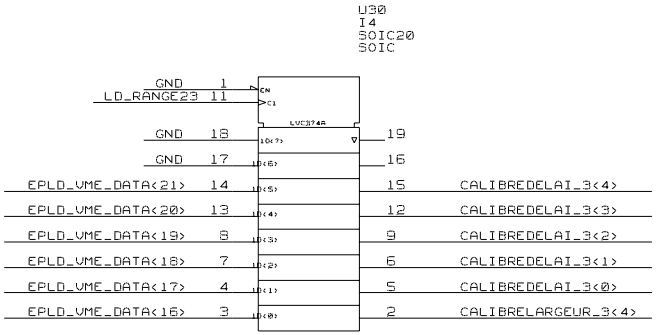
REF:	
DAC	ETUDE:O. BOURRION — PAGE:
	DESSIN:CARTE TEMPO — DATE:
	PAGE 6

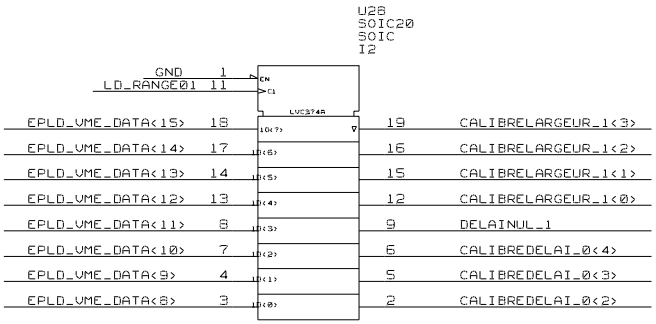

CONVMEP1
I13
J1
96
D15-0
A23-1
AM5-0
LWORD
AS
DS1
DS0
WRITE
DTACK
BERR
IRQ7
IRQ6
IRQ5
IRQ4
IRQ3
IRQ2
IRQ1
IACK
IACKIN
IACKOUT
BR3
BR2
BR1
BR0
BG3IN
BG3OUT
BG2IN
BG2OUT
BG1IN
BG1OUT
BG0IN
BG0OUT
BBSY
BCLR
SYSCLK
SYSRESET
SYSFAIL
ACFAIL
SERDAT
SERCLK
+5
+SSTDBY
+12
-12
0v
BUS_VME_DATA<15,,0>
BUS_VME_ADD<23,,1>
BUS_VME_AM<5,,0>
BUS_VME_SYSCLK
BUS_VME_SYSRESET
5V
-12V
GND
I19
I18
I16
I15
I14
I12
C24
C25
C26
C27
C28
C29
47U
MV16V_20%
I20
J2
CONVME
P2
D31-16
A31-24
E/S A32-1
E/S C32-1
Reserve
BUS_VME_DATA<31,,16>
BUS_VME_ADD<31,,24>
C30
I27
100N
C80SSS0V10%
C31
I28
3V3
I1
U19
TSSOP
IDT74FCT16424ST
I2
I3
U20
I4
DIR
OE*
GND
3V3
EPLD_VME_WRITE
OE_BUS
BUS_VME_DATA<11>
BUS_VME_DATA<3>
BUS_VME_DATA<10>
BUS_VME_DATA<2>
BUS_VME_DATA<9>
BUS_VME_DATA<1>
BUS_VME_DATA<8>
BUS_VME_DATA<0>
EPLD_VME_DATA<11>
EPLD_VME_DATA<3>
EPLD_VME_DATA<10>
EPLD_VME_DATA<2>
EPLD_VME_DATA<9>
EPLD_VME_DATA<1>
EPLD_VME_DATA<8>
EPLD_VME_DATA<0>
BUS_VME_DATA<15>
BUS_VME_DATA<7>
BUS_VME_DATA<14>
BUS_VME_DATA<6>
BUS_VME_DATA<13>
BUS_VME_DATA<5>
BUS_VME_DATA<12>
BUS_VME_DATA<4>
EPLD_VME_DATA<15>
EPLD_VME_DATA<7>
EPLD_VME_DATA<14>
EPLD_VME_DATA<6>
EPLD_VME_DATA<13>
EPLD_VME_DATA<5>
EPLD_VME_DATA<12>
EPLD_VME_DATA<4>
BUS_VME_IACKIN
BUS_VME_IACK
BUS_VME_AS
BUS_VME_WRITE
BUS_VME_AM<5>
BUS_VME_DS0
BUS_VME_SYSRESET
BUS_VME_SYSCLK
EPLD_VME_IACKIN
EPLD_VME_IACK
EPLD_VME_AS
EPLD_VME_WRITE
EPLD_VME_AM<5>
EPLD_VME_DS0
EPLD_VME_SYSRESET
EPLD_VME_SYSCLK
BUS_VME_ADD<1>
BUS_VME_ADD<2>
BUS_VME_ADD<3>
BUS_VME_ADD<4>
BUS_VME_ADD<5>
BUS_VME_ADD<6>
BUS_VME_AM<4>
EPLD_VME_IACKOUT
EPLD_VME_ADD<1>
EPLD_VME_ADD<2>
EPLD_VME_ADD<3>
EPLD_VME_ADD<4>
EPLD_VME_ADD<5>
EPLD_VME_ADD<6>
EPLD_VME_AM<4>
BUS_VME_IACKOUT
REF:
INTERFACE VME 1
ETUDE: O. BOURRION
DESSIN: CARTE TEMPO
PAGE:
DATE:
PAGE 7

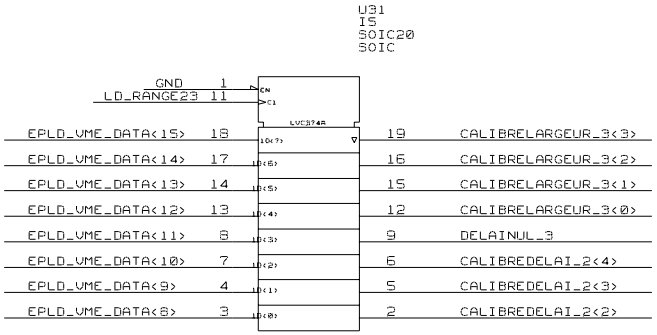

I20
U21
TSSOP
IDT74FCT164245T
DIR
OE*
EPLD_VME_WRITE
OE_BUS
BUS_VME_DATA(31)
BUS_VME_DATA(30)
BUS_VME_DATA(29)
BUS_VME_DATA(28)
BUS_VME_DATA(27)
BUS_VME_DATA(26)
BUS_VME_DATA(25)
BUS_VME_DATA(24)
EPLD_VME_DATA(31)
EPLD_VME_DATA(30)
EPLD_VME_DATA(29)
EPLD_VME_DATA(28)
EPLD_VME_DATA(27)
EPLD_VME_DATA(26)
EPLD_VME_DATA(25)
EPLD_VME_DATA(24)
GND
4, 10, 15
21, 28, 34
39, 45
3V3
31, 42
5V
7, 18
I31
U21
TSSOP
IDT74FCT164245T
BUS_VME_DATA(23)
BUS_VME_DATA(22)
BUS_VME_DATA(21)
BUS_VME_DATA(20)
BUS_VME_DATA(19)
BUS_VME_DATA(18)
BUS_VME_DATA(17)
BUS_VME_DATA(16)
EPLD_VME_DATA(23)
EPLD_VME_DATA(22)
EPLD_VME_DATA(21)
EPLD_VME_DATA(20)
EPLD_VME_DATA(19)
EPLD_VME_DATA(18)
EPLD_VME_DATA(17)
EPLD_VME_DATA(16)
I18
U22
SO14
BUS_VME_IRQ(7)
EPLD_IRQ(7)
I19
U22
BUS_VME_IRQ(6)
EPLD_IRQ(6)
I21
U22
BUS_VME_IRQ(5)
EPLD_IRQ(5)
I22
U22
BUS_VME_IRQ(4)
EPLD_IRQ(4)
I23
U22
BUS_VME_IRQ(3)
EPLD_IRQ(3)
I24
U22
BUS_VME_IRQ(2)
EPLD_IRQ(2)
I25
U23
BUS_VME_IRQ(1)
EPLD_IRQ(1)
R72 I45 1K
R72 I44 1K
R72 I43 1K
R72 I42 1K
R72 I41 1K
R72 I40 1K
R72 I39 1K
I26
U23
BUS_VME_DTACK
EPLD_DTACK
R72 I38 1K
I27
U23
I28
U23
I29
U23
I30
U23
C32 I33 100N C805S50V10%
C35 I34 100N C805S50V10%
C33 I35 100N C805S50V10%
C34 I37 100N C805S50V10%
C36 I36 100N C805S50V10%
I4
S2
PT65103
I5
S3
PT65103
R73 I1 1K
I2
I3
I6
U24
DIP20
74AC521
IEG*
OEG*
BUS_VME_ADD(31)
BUS_VME_ADD(30)
BUS_VME_ADD(29)
BUS_VME_ADD(28)
BUS_VME_ADD(27)
BUS_VME_ADD(26)
BUS_VME_ADD(25)
BUS_VME_ADD(24)
19 BUS_A31_A24_OK
I12
S4
PT65103
I13
S5
PT65103
R74 I7 1K
I10
I11
I8
U25
DIP20
BUS_VME_ADD(23)
BUS_VME_ADD(22)
BUS_VME_ADD(21)
BUS_VME_ADD(20)
BUS_VME_ADD(19)
BUS_VME_ADD(18)
BUS_VME_ADD(17)
BUS_VME_ADD(16)
19 BUS_A23_A8_OK
I9
U26
DIP20
AM(3..0)=1001
ACCES AUX DONNEES EN
MODE UTILISATEUR
STANDARD OU ETENDU
LWORD MODE 32 BIT UNIQUEMENT
BUS_VME_AM(3)
BUS_VME_AM(2)
BUS_VME_AM(1)
BUS_VME_AM(0)
BUS_VME_LWORD
BUS_VME_ADD(7)
BUS_VME_DS1
19 BUS_MODE_OK
REF: CARTE TEMPO
INTERFACE VME 2
ETUDE:
PAGE:
DESSIN:O. BOURRION
DATE:
PAGE 8

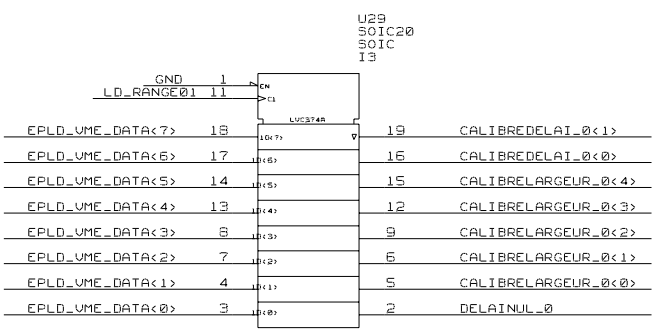

U1
-5
I1
XC2S30PQ208
TDI_XILINX 159
TCK_XILINX 207
TMS_XILINX 2
DCLK 155
PROGRAM 106
SLAVE SERIAL MODE,
WITHOUT PULL UPS
3V3 54
3V3 50
3V3 52
VALIDREJECTCLK
VALIDEVENTCLK
EPLD_VME_SYSCLK 185
VALIDEVENTOUT 3
SV TOL
EVENTTTL 4
SV TOL
TEMPSMORTTTL 8
SV TOL
RAZINTTL 9
LED_EVENT 15
LED_TEMPSMORT 16
VALIDTTL 18
RAZOUTTTL 22
RAZINTERNETTL 23
LD_RANGE01 42
LD_RANGE23 43
SV TOL
REJECTTTL 62
VALIDREJECTOUT 61
NLD_DAC01 69
NLD_DAC23 70
EPLD_VME_DATA<31> 74
EPLD_VME_DATA<30> 75
EPLD_VME_DATA<29> 81
EPLD_VME_DATA<28> 82
EPLD_VME_DATA<27> 83
EPLD_VME_DATA<26> 86
EPLD_VME_DATA<25> 87
EPLD_VME_DATA<24> 88
EPLD_VME_DATA<23> 89
EPLD_VME_DATA<22> 90
EPLD_VME_DATA<21> 94
EPLD_VME_DATA<20> 95
EPLD_VME_DATA<19> 96
EPLD_VME_DATA<18> 99
EPLD_VME_DATA<17> 100
EPLD_VME_DATA<16> 101
SV TOL
BUS_A31_A24_OK 102
TDO 157
DONE 104
INIT 107
TOGGLE 1
P1
TEST
I7
OE_BUS 114
EPLD_IRQ<1> 119
EPLD_IRQ<2> 120
EPLD_IRQ<3> 121
EPLD_IRQ<4> 122
EPLD_IRQ<5> 123
EPLD_IRQ<6> 127
EPLD_IRQ<7> 138
EPLD_VME_ADD<1> 139
EPLD_VME_ADD<2> 140
EPLD_VME_ADD<3> 141
EPLD_VME_ADD<4> 147
EPLD_VME_ADD<5> 148
EPLD_VME_ADD<6> 151
BUS_MODE_OK 152
SV TOL
D 153
SV TOL
BUS_A23_A8_OK 160
EPLD_VME_AM<4> 161
EPLD_VME_IACKOUT 162
EPLD_VME_IACKIN 163
EPLD_VME_IACK 166
EPLD_VME_AS 167
EPLD_DTACK 168
EPLD_VME_AM<5> 172
EPLD_VME_WRITE 173
EPLD_VME_DS0 174
RESET_OUT 175
EPLD_VME_DATA<15> 176
EPLD_VME_DATA<7> 179
EPLD_VME_DATA<14> 180
EPLD_VME_DATA<6> 181
EPLD_VME_DATA<13> 187
EPLD_VME_DATA<5> 191
EPLD_VME_DATA<12> 192
EPLD_VME_DATA<4> 193
EPLD_VME_DATA<11> 194
EPLD_VME_DATA<3> 195
EPLD_VME_DATA<10> 199
EPLD_VME_DATA<2> 200
EPLD_VME_DATA<9> 201
EPLD_VME_DATA<1> 204
EPLD_VME_DATA<8> 205
EPLD_VME_DATA<0> 206
3V3
12 25 39 53 65 78 92 105
117 130 144 156 170 184 197 208
GND
1,11,19 25,32,40 51,64,72 79,85,93 103,115,124 131,137,145 158,169,177 183,190,198
2V5
28,66,76 91,120,171 186,196
I20
ISPFLASH
EPLD_VME_SYSCLK CLK
EPLD_VME_SYSRESET NRST
D
DCLK
RESET_OUT
PROGRAM
INIT
DONE
3V3 3V3
GND GND
R75 I30 3.32K S125MW1%
R76 I31 2.15K S125MW1%
R77 I32 2.15K S125MW1%
R78 I33 2.15K S125MW1%
R79 I34 2.15K S125MW1%
R80 I29 1.15K S125MW1%
Q10 I26 SOT23 GSD
D7 I27 DEFAULT ROUGE
R81 I28 105 S125MW1%
D8 I22 DEFAULT VERTE
R85 I21 105 S125MW1%
R82 I24 1.15K S125MW1%
Q11 I23 GSD BSS138
I3 R84 1.15K S125MW1%
R83 I36 1.15K S125MW1%
R86 I35 1.15K S125MW1%
C39 I9 100N C805550V10%
C41 I10 100N C805550V10%
C43 I11 100N C805550V10%
C45 I25 100N C805550V10%
C37 I18 100N C805550V10%
C38 I17 100N C805550V10%
C40 I16 100N C805550V10%
C42 I15 100N C805550V10%
C44 I14 100N C805550V10%
REF:
FPGA VME
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:
PAGE 9

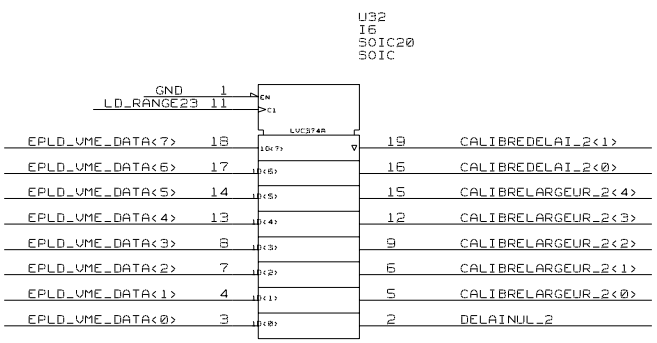




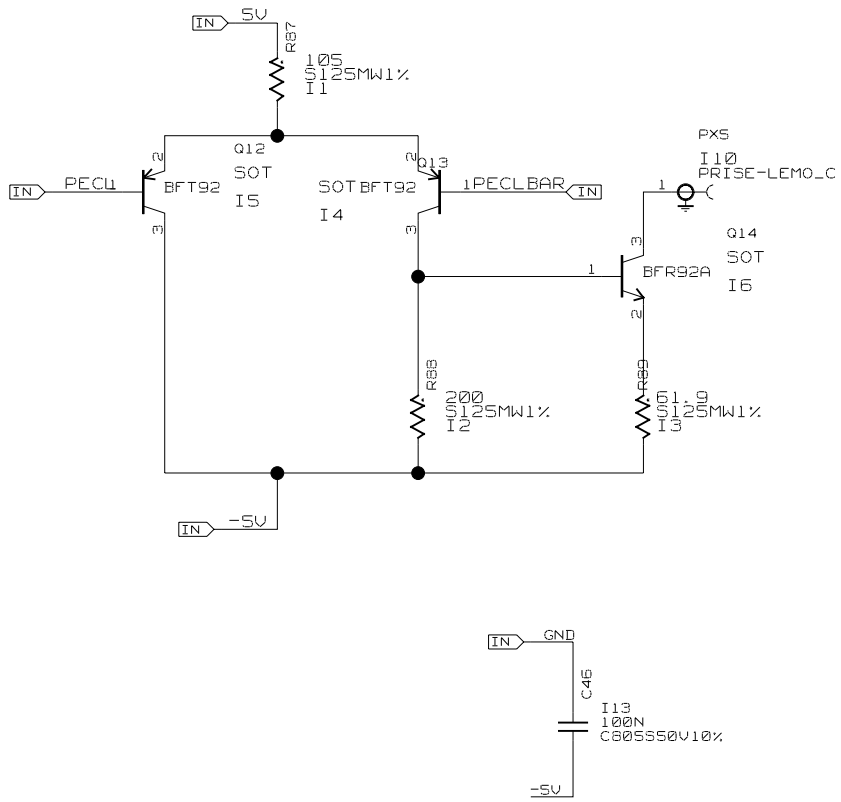






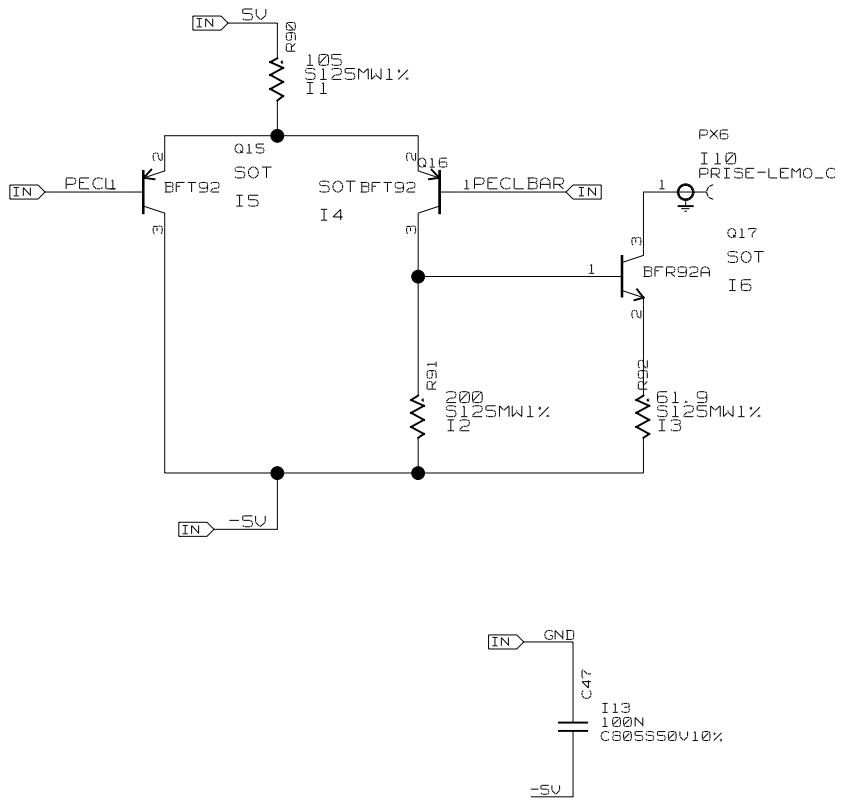


REF:		
RANGE_CTRL	ETUDE:	PAGE:
	DESSIN:	DATE:
	PAGE 10	



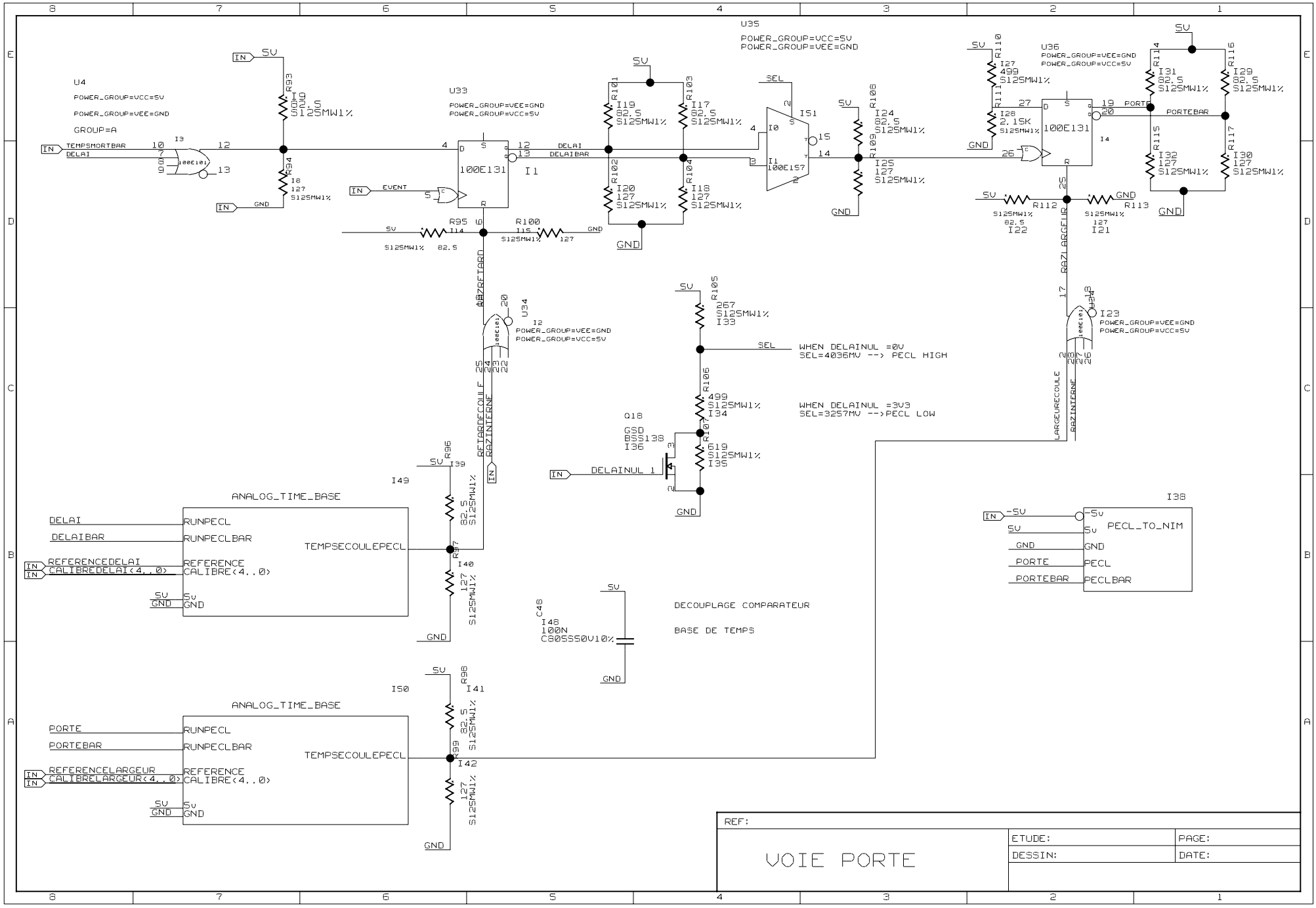
REF: PECL TO NIM

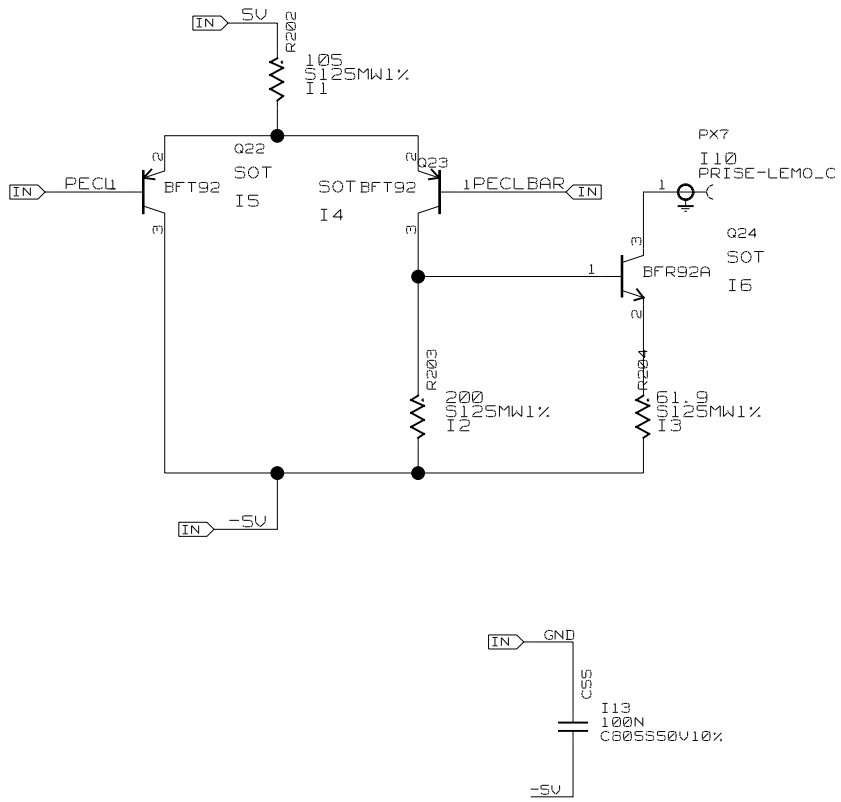
	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:



REF: PECL TO NIM

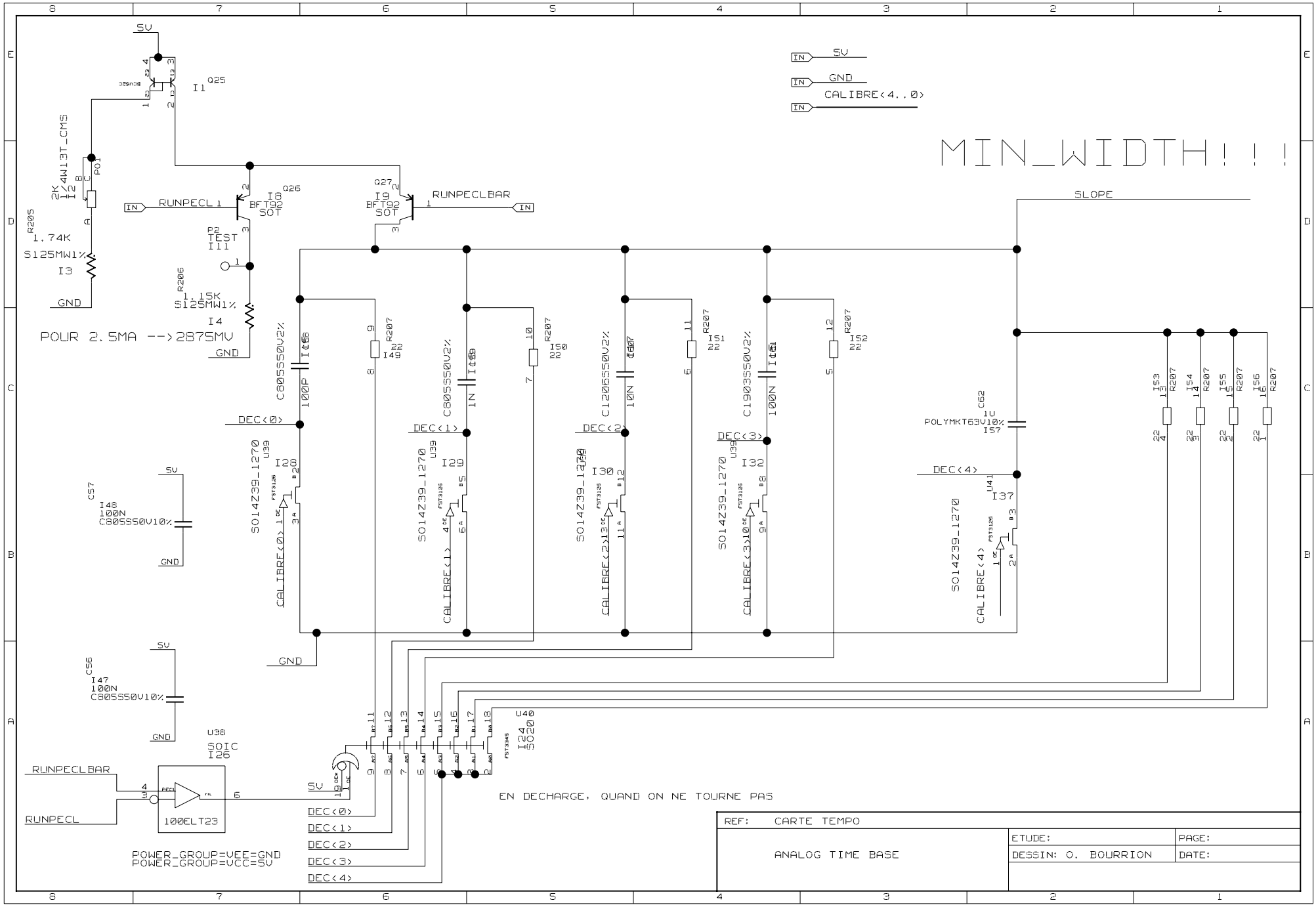
	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:

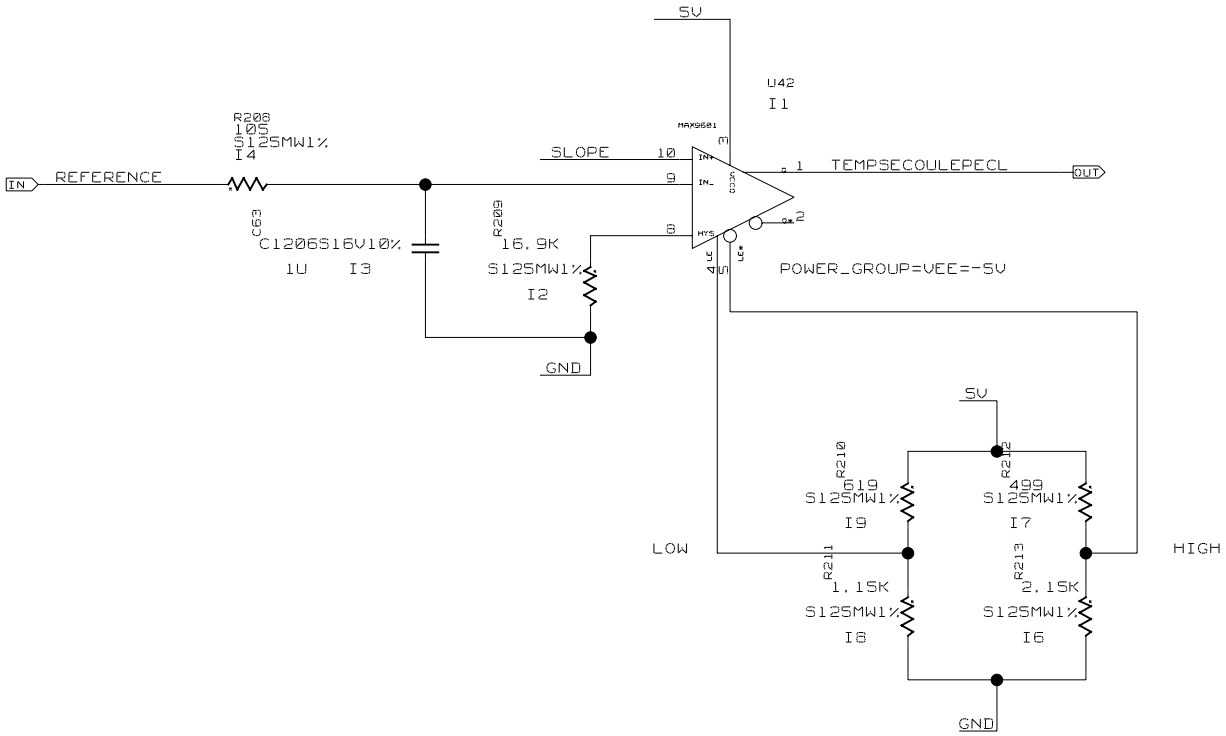

U4
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
GROUP=A
TEMPSMORTBAR
DELAI
U33
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
100E131
EVENT
DELAI
DELAIBAR
U35
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
100E157
U36
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
PORTE
PORTEBAR
U34
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
WHEN DELAINUL =0V
SEL=4036MV --> PECL HIGH
WHEN DELAINUL =3V3
SEL=3257MV -->PECL LOW
DELAINUL
ANALOG_TIME_BASE
RUNPECL
RUNPECLBAR
REFERENCE
CALIBRE<4..0>
TEMPSECOULEPECL
REFERENCEDELAI
CALIBREDELAI<4..0>
REFERENCELARGEUR
CALIBRELARGEUR<4..0>
PECL_TO_NIM
PECL
PECLBAR
DECOUPLAGE COMPARATEUR
BASE DE TEMPS
REF:
VOIE PORTE
ETUDE:
PAGE:
DESSIN:
DATE:



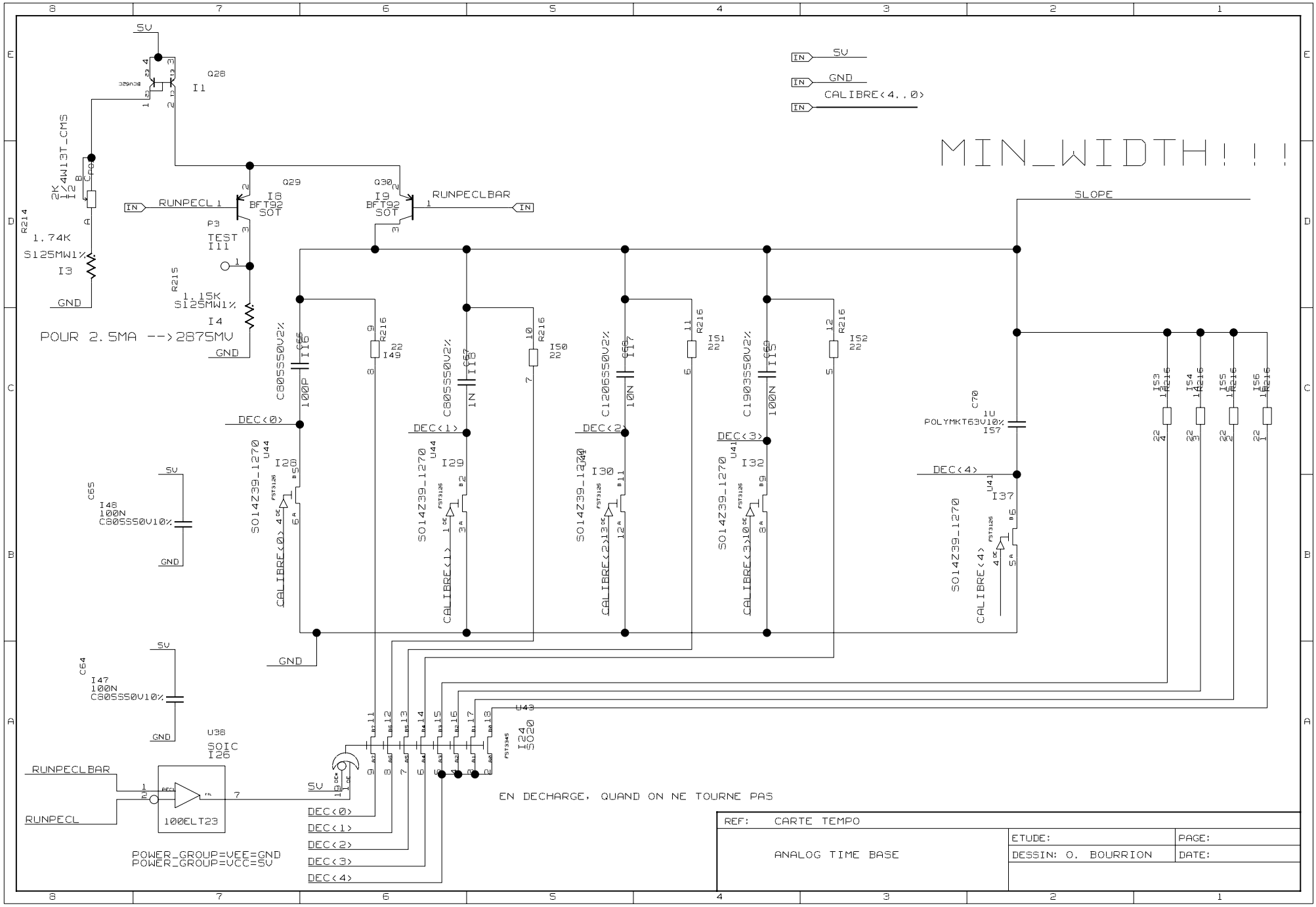
REF: PECL TO NIM

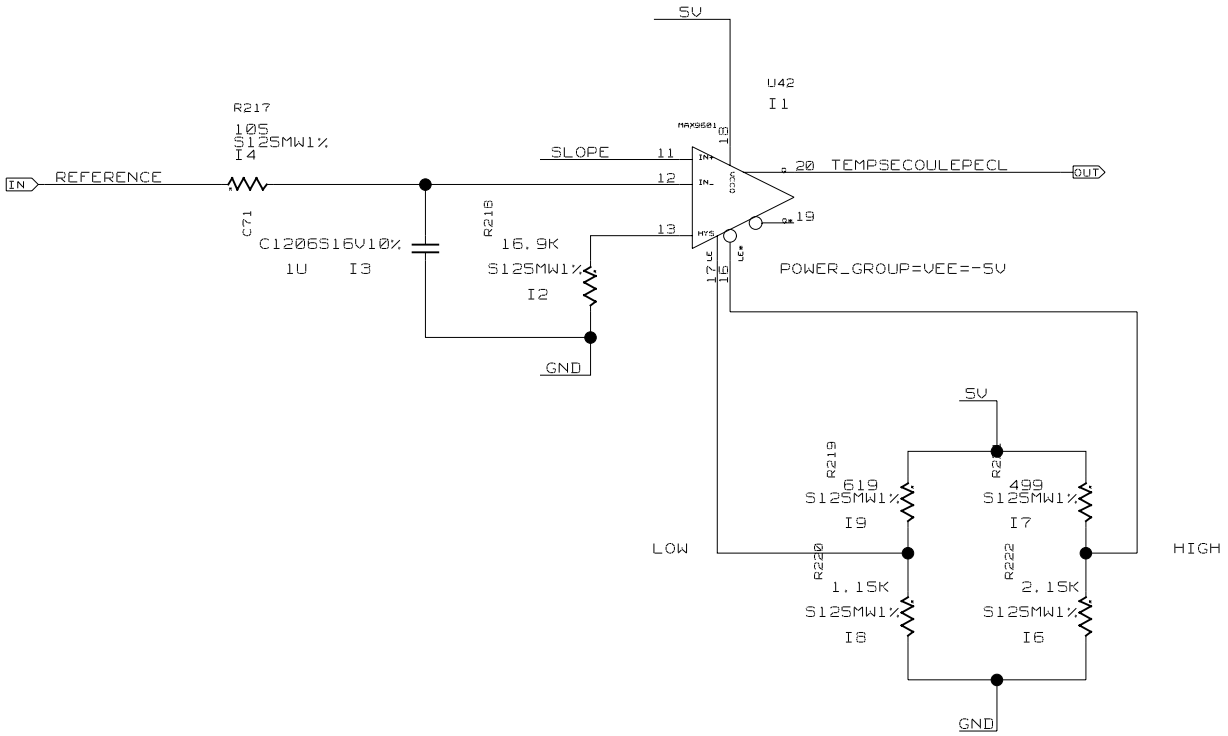
	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!
SLOPE
RUNPECL
RUNPECLBAR
CALIBRE<4..0>
DEC<0>
DEC<1>
DEC<2>
DEC<3>
DEC<4>
POUR 2.5MA -->2875MV
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:

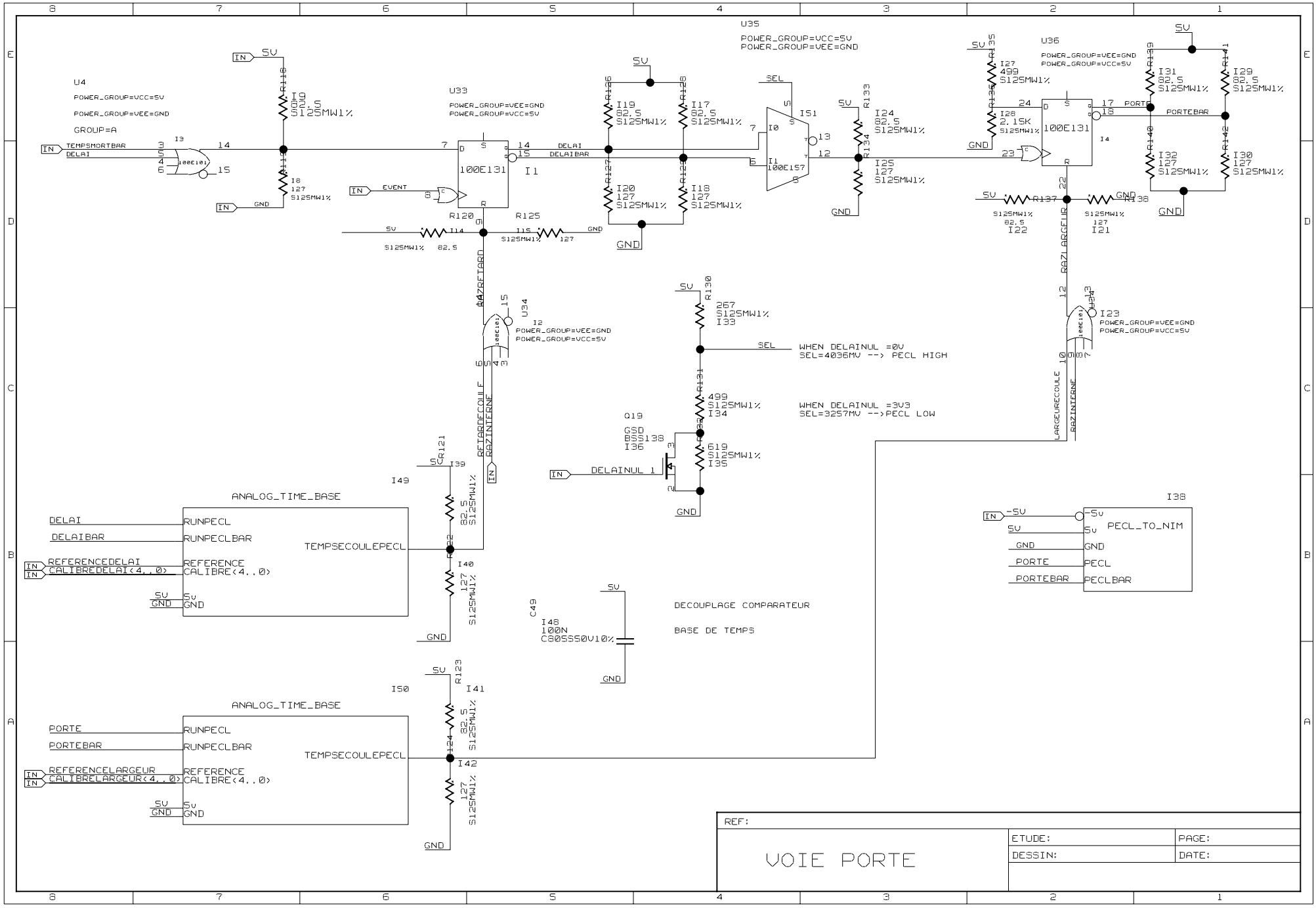


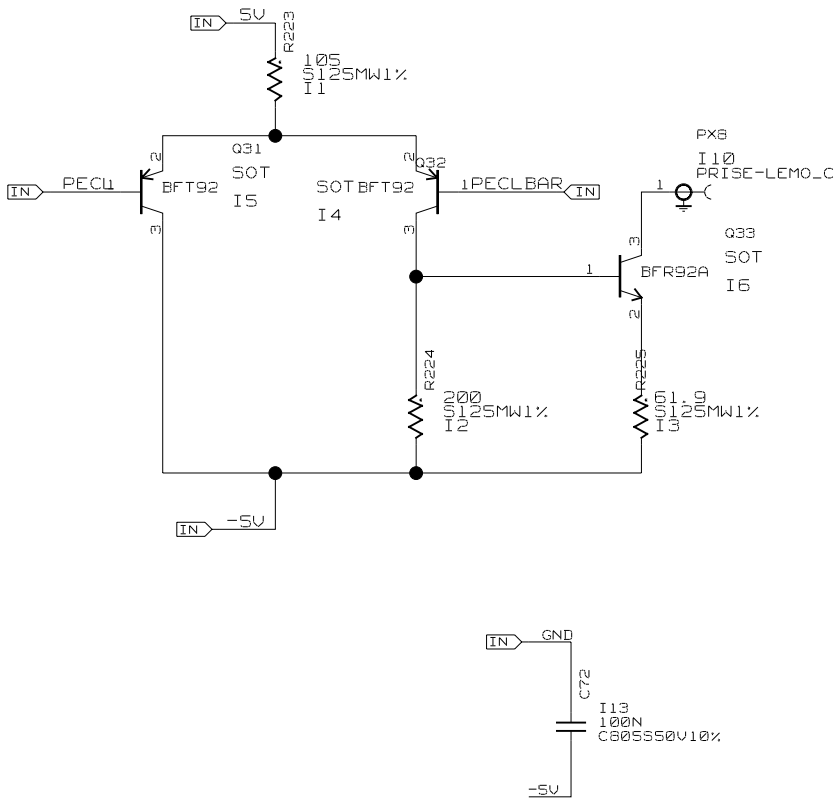
REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!
5V
GND
CALIBRE<4..0>
SLOPE
RUNPECL
RUNPECLBAR
Q28
Q29
Q30
BFT92
SOT
TEST
POUR 2.5MA -->2875MV
R214
1.74K
S125MW1%
R215
1.15K
S125MW1%
2K
4W13T_CMS
C80S550V2%
100P
1N
10N
100N
C190350V2%
C70
1U
POLYMKT63V10%
DEC<0>
DEC<1>
DEC<2>
DEC<3>
DEC<4>
SO14Z39_1270
CALIBRE<0>
CALIBRE<1>
CALIBRE<2>
CALIBRE<3>
CALIBRE<4>
C65
C64
100N
C80S550V10%
U38
SOIC
100ELT23
U43
SO20
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:



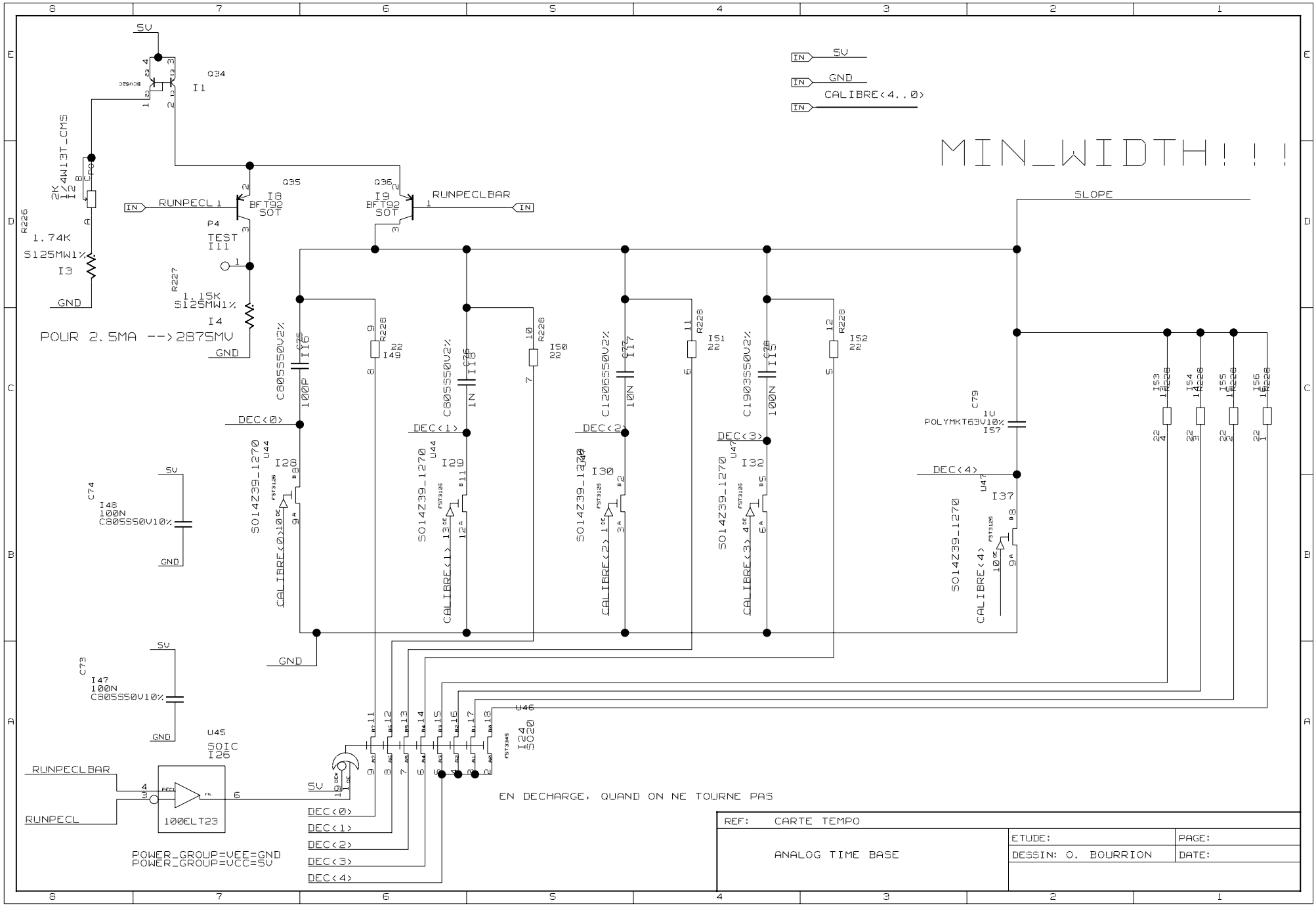
REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:

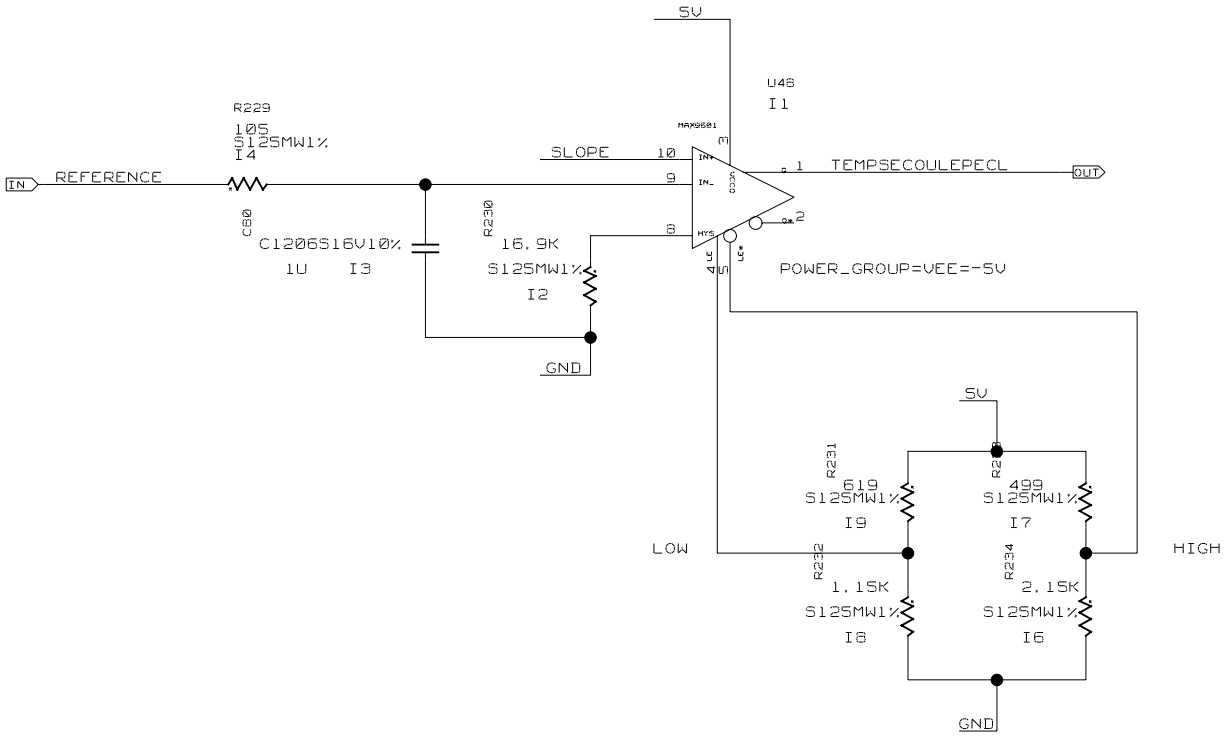

U4
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
GROUP=A
TEMPSMORTBAR
DELAI
U33
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
100E131
EVENT
DELAI
DELAIBAR
U35
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
SEL
U36
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
PORTE
PORTEBAR
U34
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
WHEN DELAINUL =0V
SEL=4036MV --> PECL HIGH
WHEN DELAINUL =3V3
SEL=3257MV -->PECL LOW
DELAINUL
ANALOG_TIME_BASE
RUNPECL
RUNPECLBAR
TEMPSECOULEPECL
REFERENCE
CALIBRE<4..0>
REFERENCEDELAI
CALIBREDELAI<4..0>
REFERENCELARGEUR
CALIBRELARGEUR<4..0>
PECL_TO_NIM
PECL
PECLBAR
DECOUPLAGE COMPARATEUR
BASE DE TEMPS
REF:
VOIE PORTE
ETUDE:
PAGE:
DESSIN:
DATE:



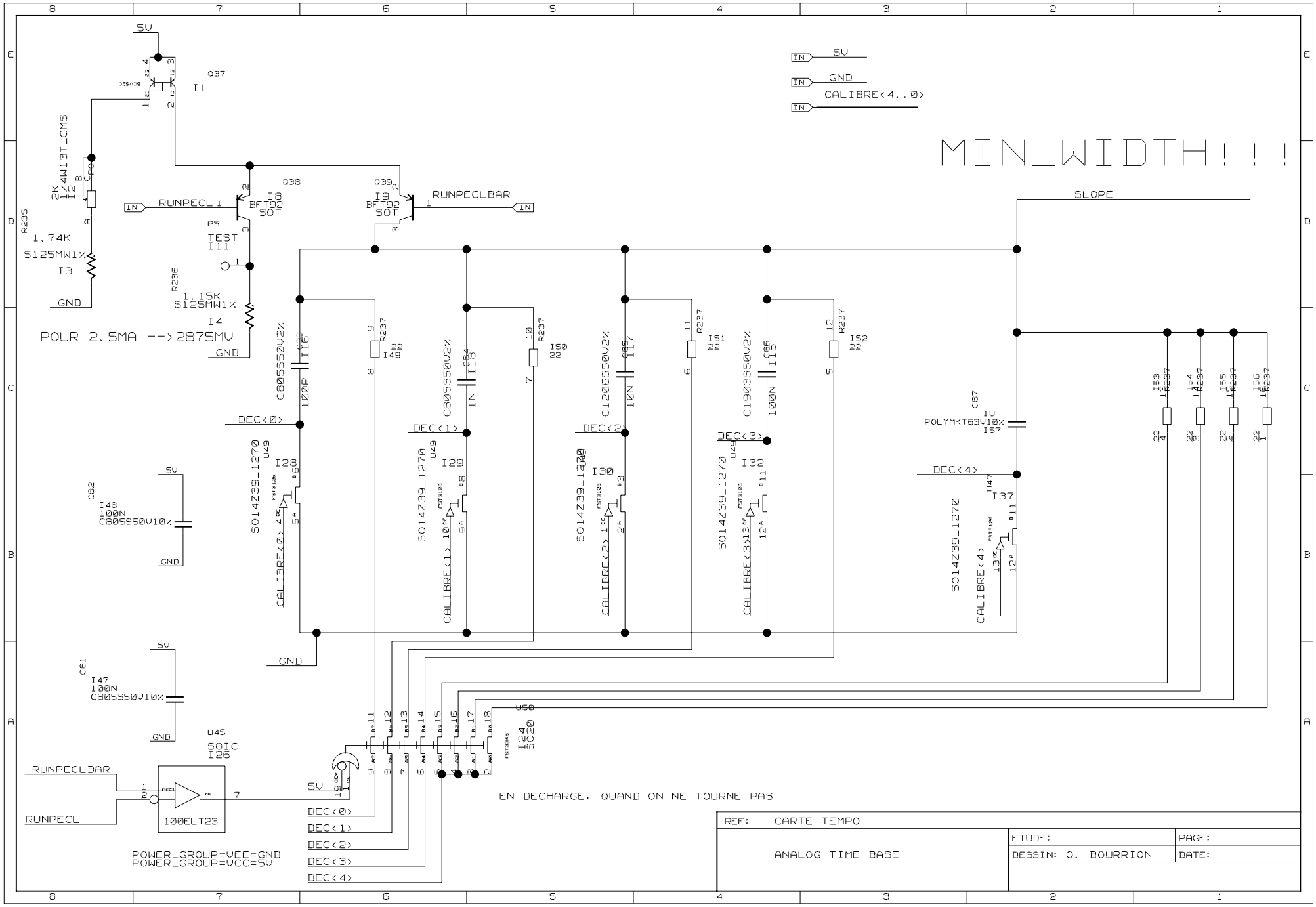
REF: PECL TO NIM

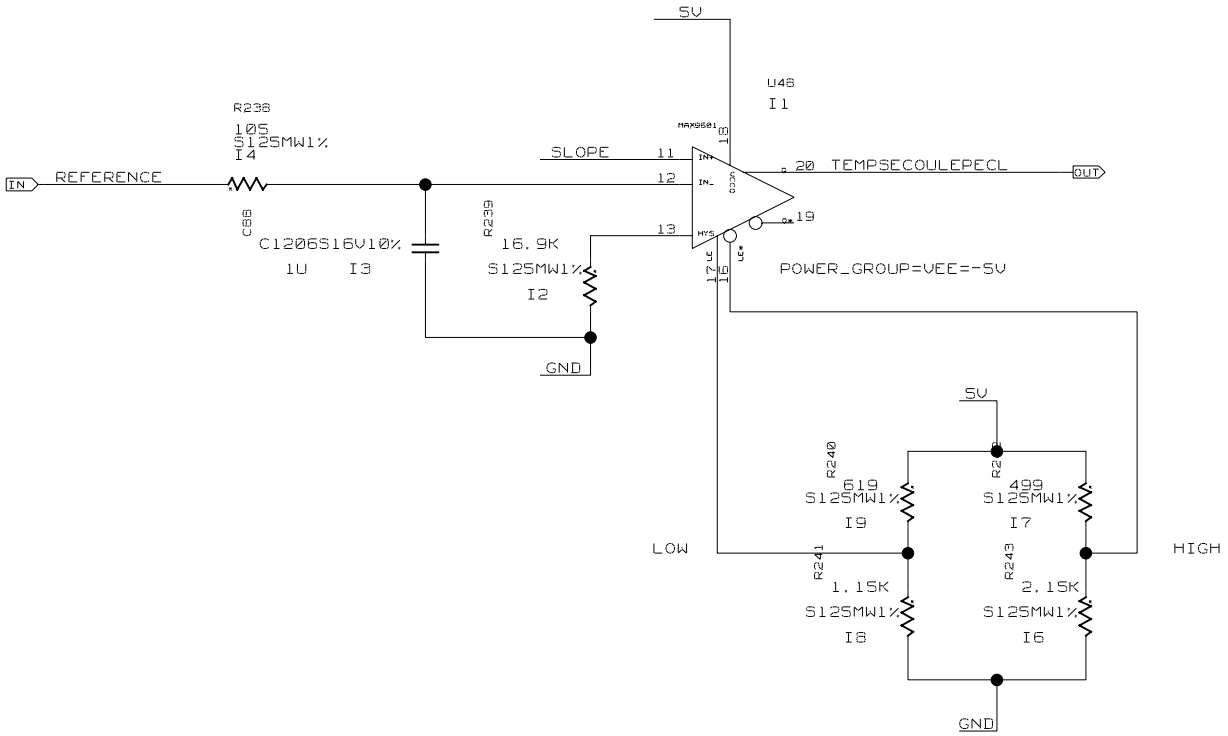
	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!
SLOPE
RUNPECL
RUNPECLBAR
CALIBRE<4..0>
POUR 2.5MA -->2875MV
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
100ELT23
SOIC
SO14Z39_1270
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:

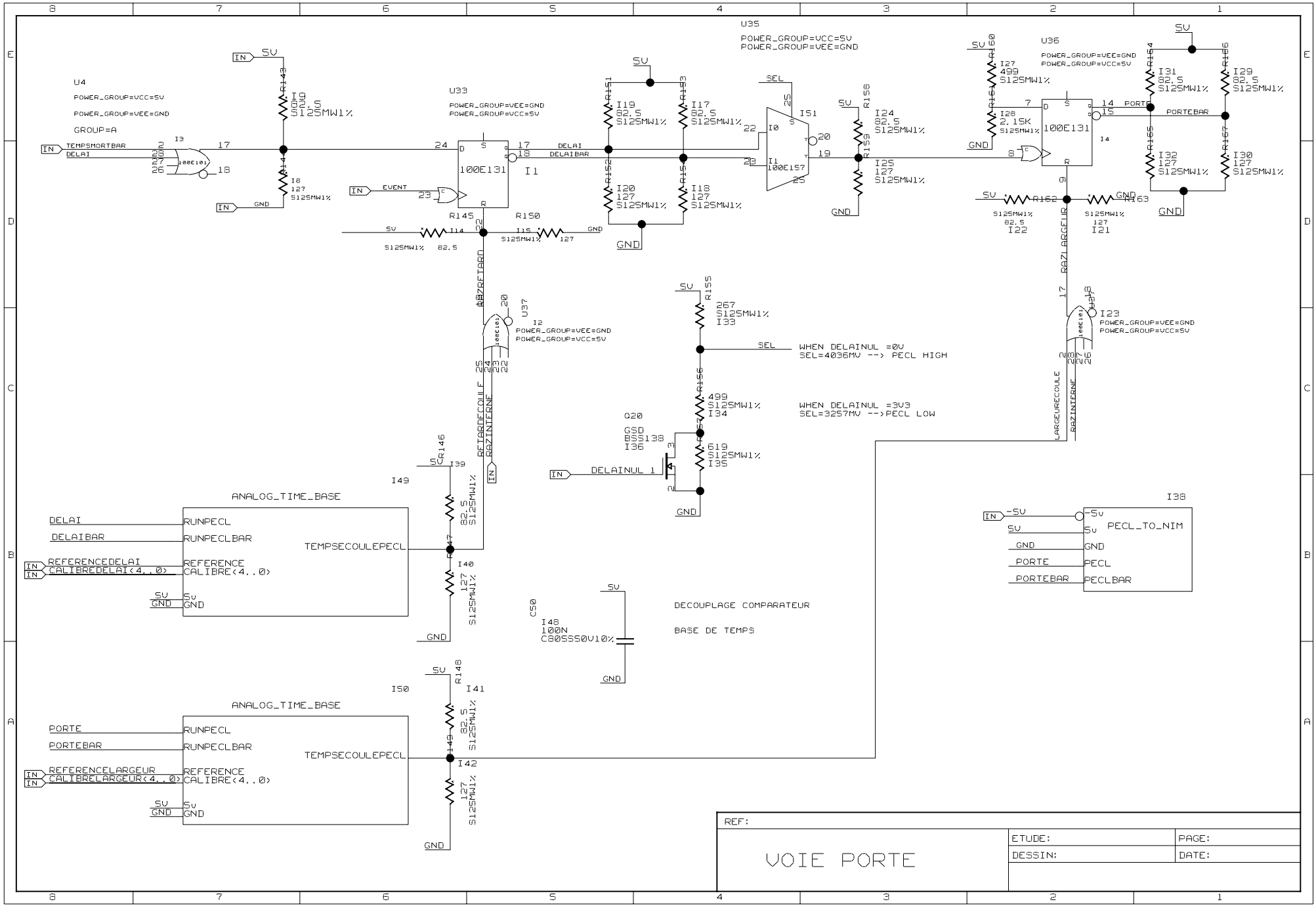


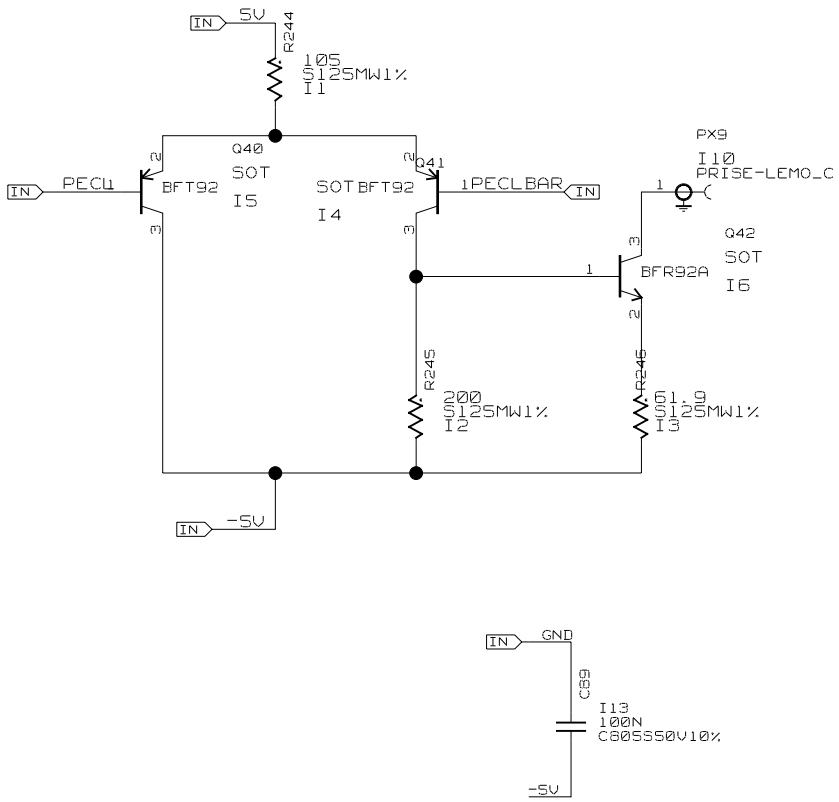
REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!
SLOPE
RUNPECL
RUNPECLBAR
POUR 2.5MA -->2875MV
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:



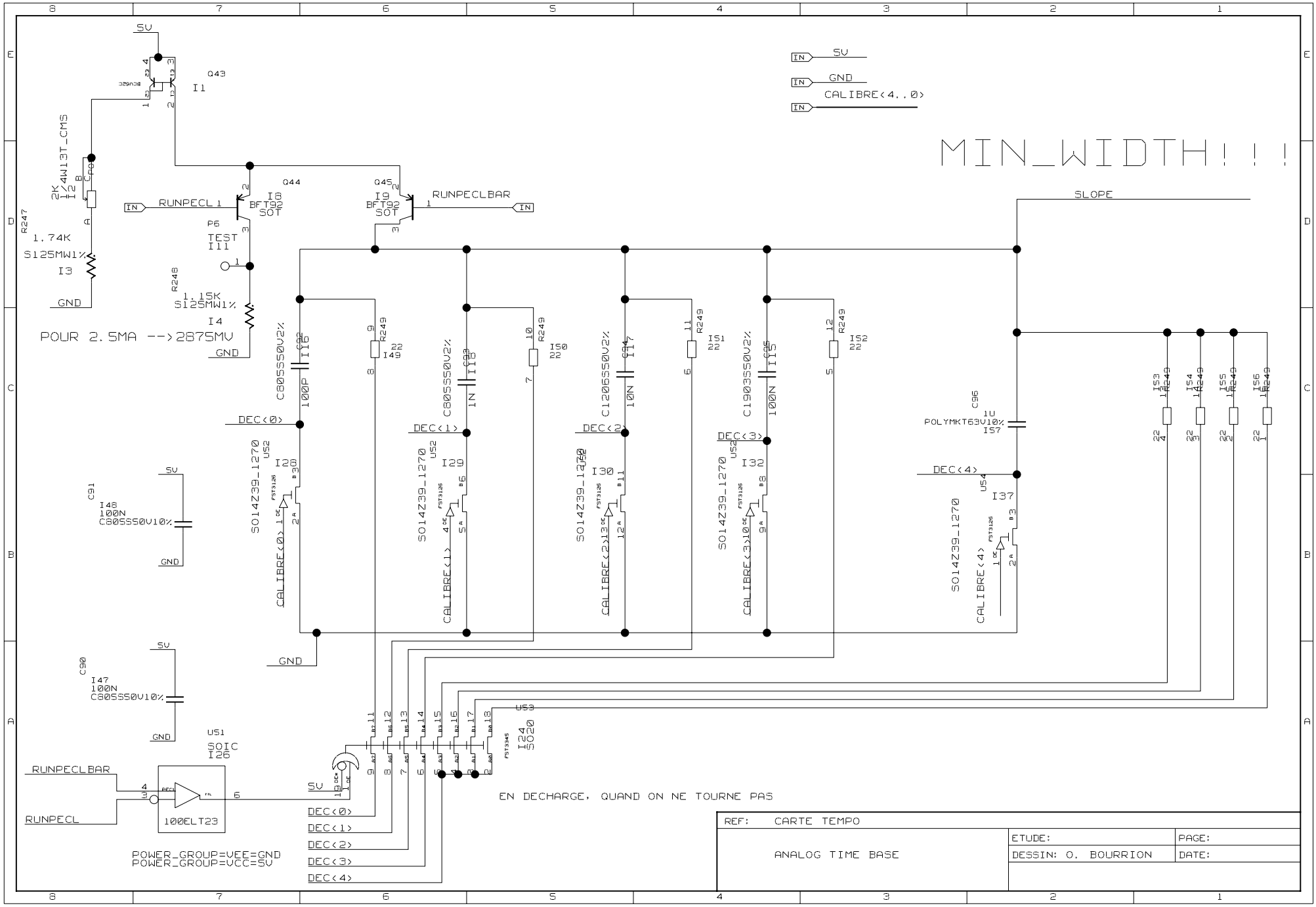
REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:

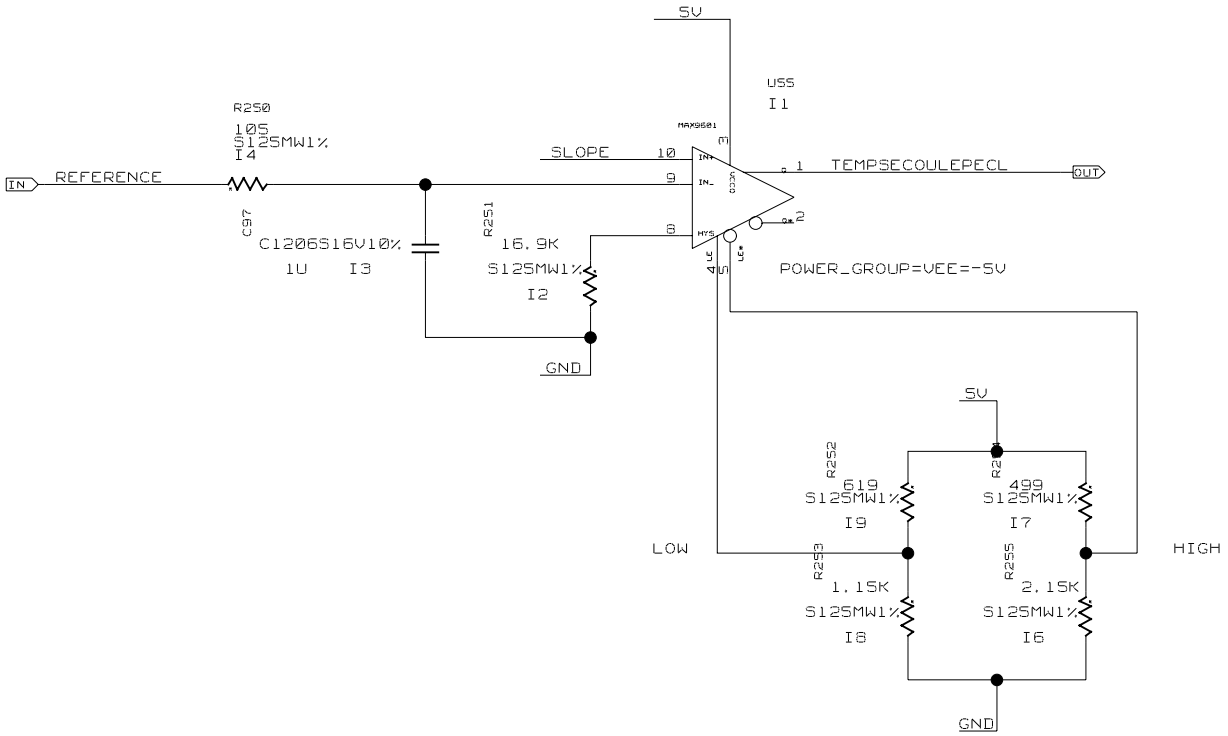

U4
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
GROUP=A
TEMPSMORTBAR
DELAI
U33
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
100E131
EVENT
DELAI
DELAIBAR
U35
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
SEL
100E157
U36
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
PORTE
PORTEBAR
U37
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
WHEN DELAINUL =0V
SEL=4036MV --> PECL HIGH
WHEN DELAINUL =3V3
SEL=3257MV -->PECL LOW
DELAINUL
ANALOG_TIME_BASE
RUNPECL
RUNPECLBAR
TEMPSECOULEPECL
REFERENCE
CALIBRE<4..0>
REFERENCEDELAI
CALIBREDELAI<4..0>
REFERENCELARGEUR
CALIBRELARGEUR<4..0>
PECL_TO_NIM
PECL
PECLBAR
DECOUPLAGE COMPARATEUR
BASE DE TEMPS
REF:
VOIE PORTE
ETUDE:
PAGE:
DESSIN:
DATE:



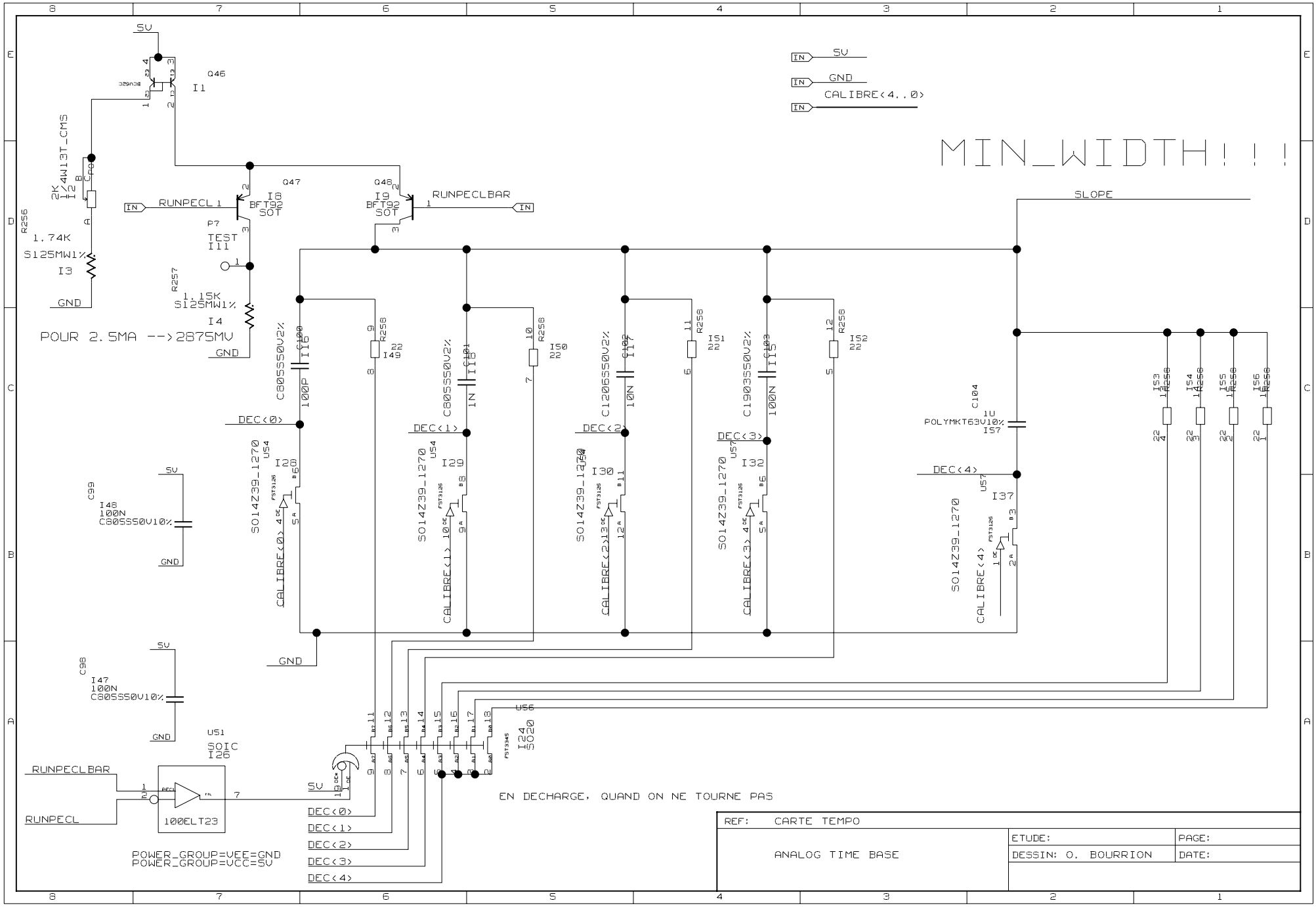
REF: PECL TO NIM

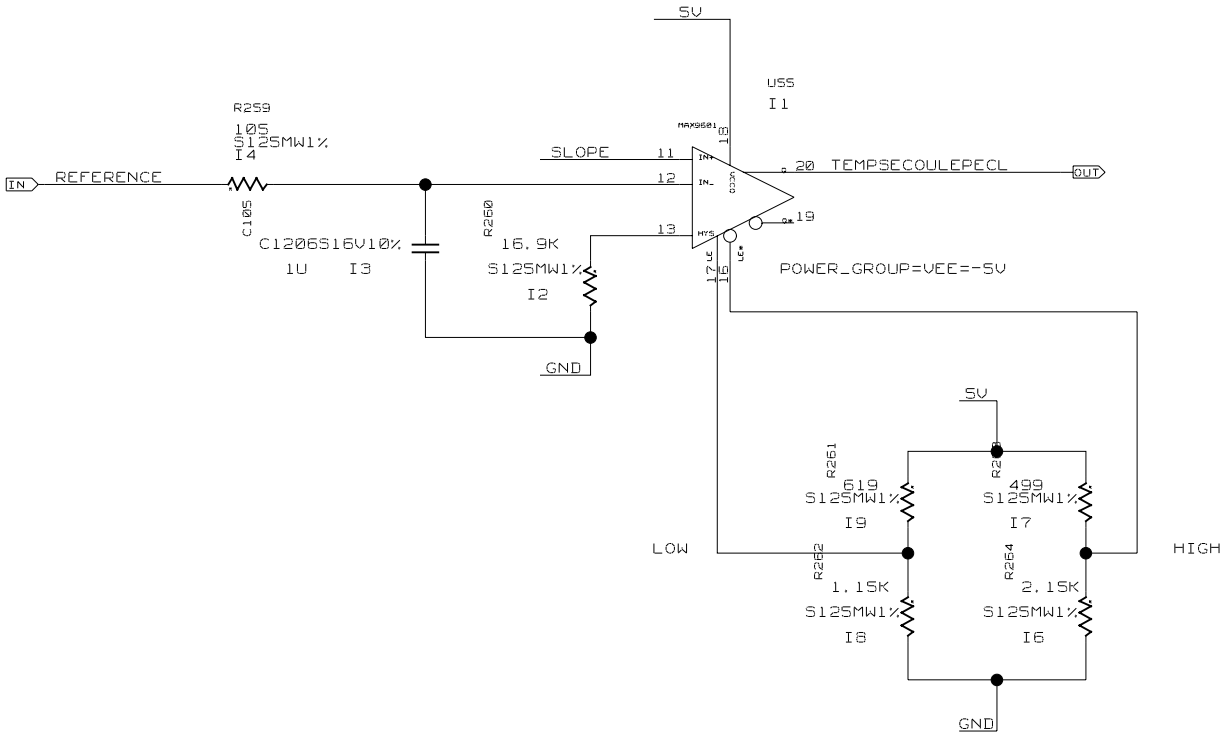
	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!
SLOPE
RUNPECL
RUNPECLBAR
CALIBRE<4..0>
5V
GND
Q43
Q44
Q45
BFT92
SOT
TEST
POUR 2.5MA -->2875MV
DEC<0>
DEC<1>
DEC<2>
DEC<3>
DEC<4>
SO14Z39_1270
CALIBRE<0>
CALIBRE<1>
CALIBRE<2>
CALIBRE<3>
CALIBRE<4>
POLYMKT63V10%
100ELT23
SOIC
SO20
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:

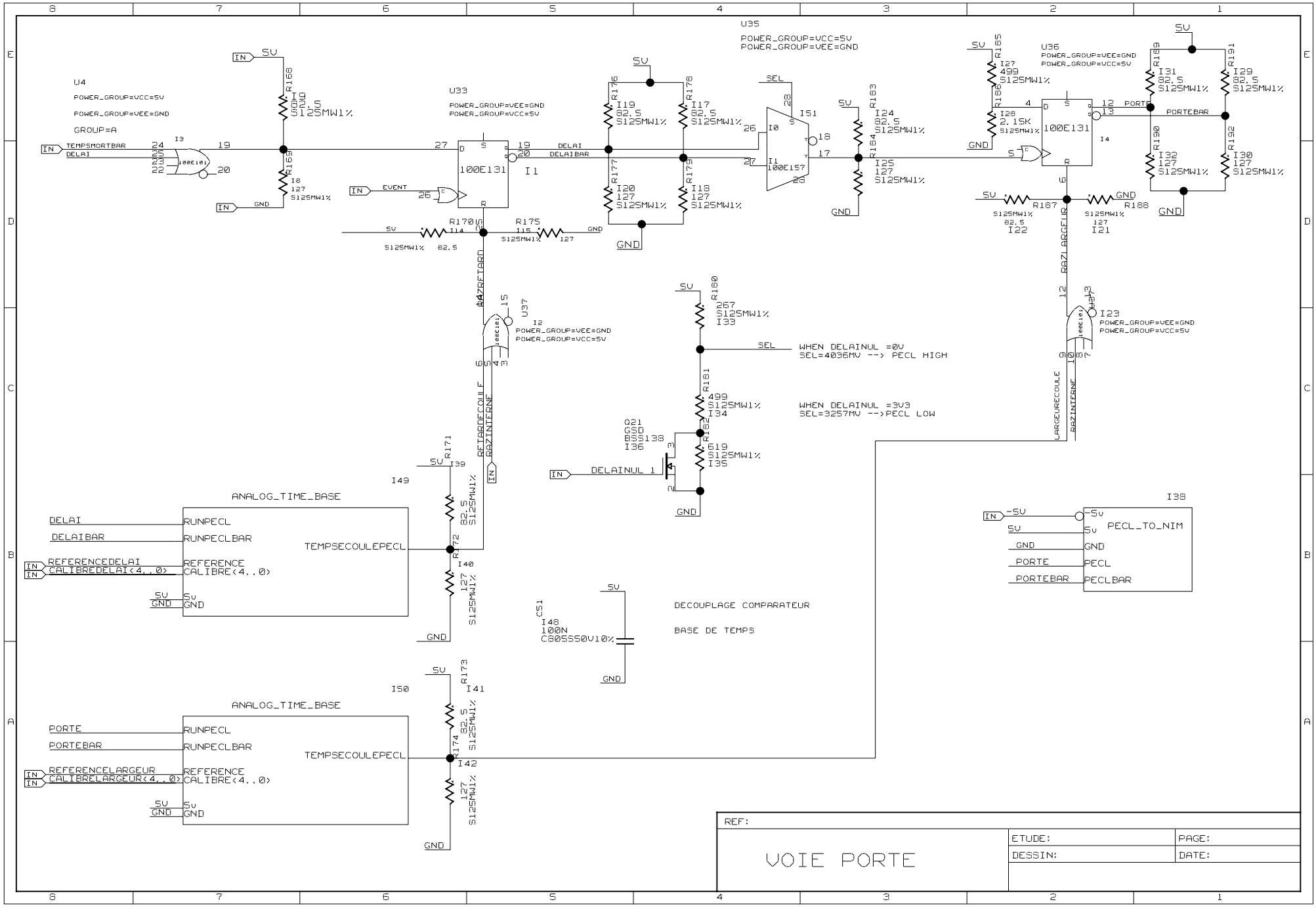


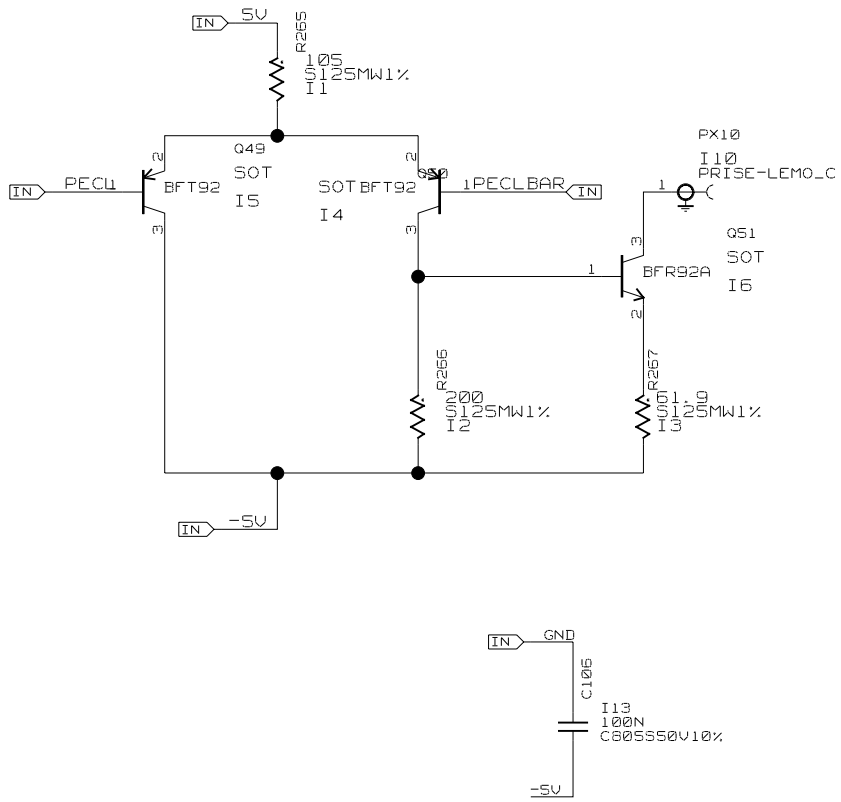
REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!!
5V
GND
CALIBRE<4..0>
SLOPE
RUNPECL
RUNPECLBAR
Q46
Q47
Q48
BFT92
SOT
TEST
POUR 2.5MA -->2875MV
1.74K
1.15K
DEC<0>
DEC<1>
DEC<2>
DEC<3>
DEC<4>
SO14Z39_1270
CALIBRE<0>
CALIBRE<1>
CALIBRE<2>
CALIBRE<3>
CALIBRE<4>
POLYMKT63V10%
100ELT23
SOIC
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:



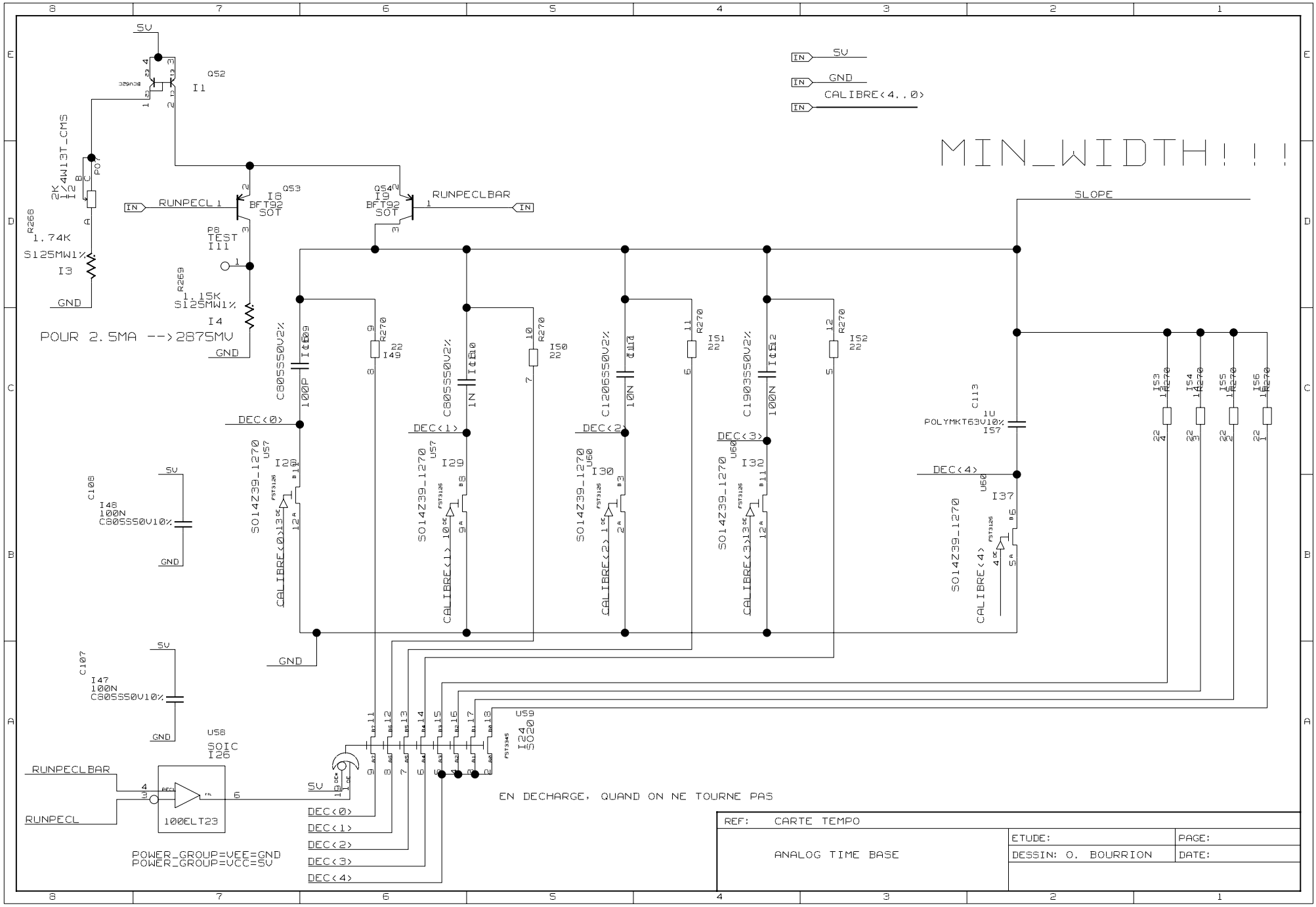
REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:

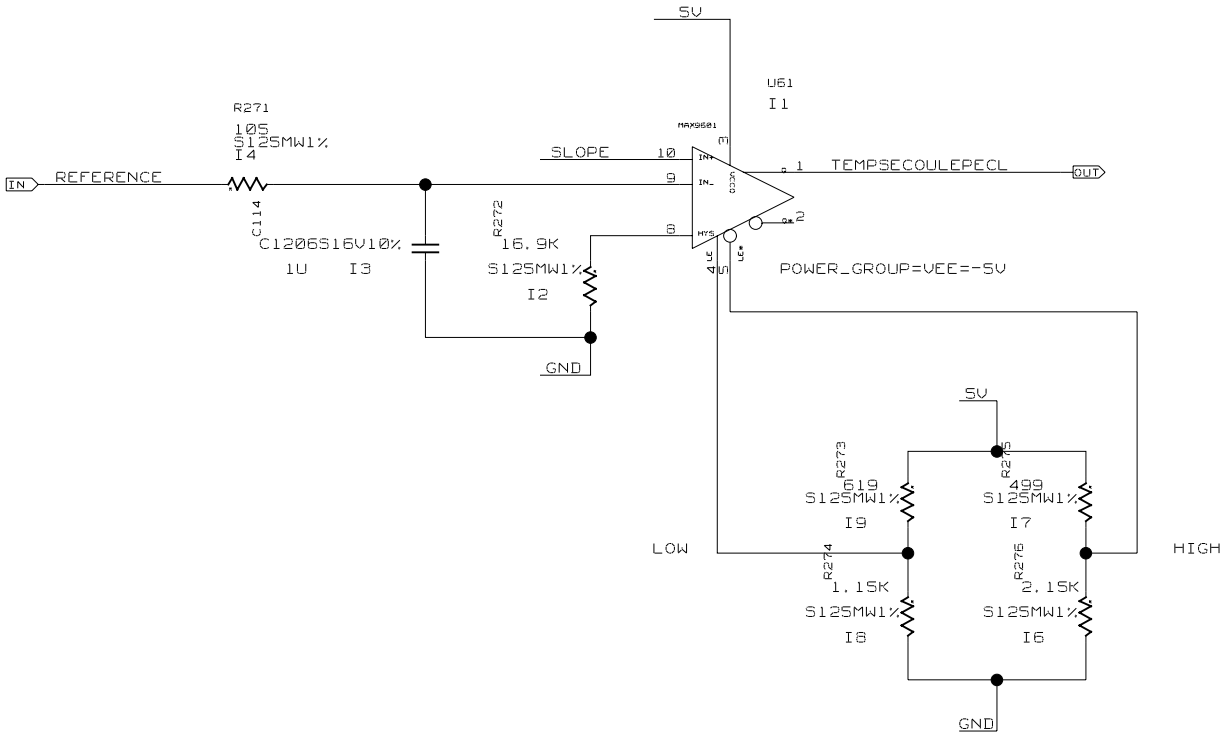

U4
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
GROUP=A
TEMPSMORTBAR
DELAI
U33
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
100E131
EVENT
DELAI
DELAIBAR
U35
POWER_GROUP=VCC=5V
POWER_GROUP=VEE=GND
SEL
100E157
U36
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
PORTE
PORTEBAR
U37
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
WHEN DELAINUL =0V
SEL=4036MV --> PECL HIGH
WHEN DELAINUL =3V3
SEL=3257MV -->PECL LOW
DELAINUL
ANALOG_TIME_BASE
RUNPECL
RUNPECLBAR
REFERENCE
CALIBRE<4..0>
TEMPSECOULEPECL
REFERENCEDELAI
CALIBREDELAI<4..0>
REFERENCELARGEUR
CALIBRELARGEUR<4..0>
DECOUPLAGE COMPARATEUR
BASE DE TEMPS
PECL_TO_NIM
PECL
PECLBAR
REF:
VOIE PORTE
ETUDE:
PAGE:
DESSIN:
DATE:



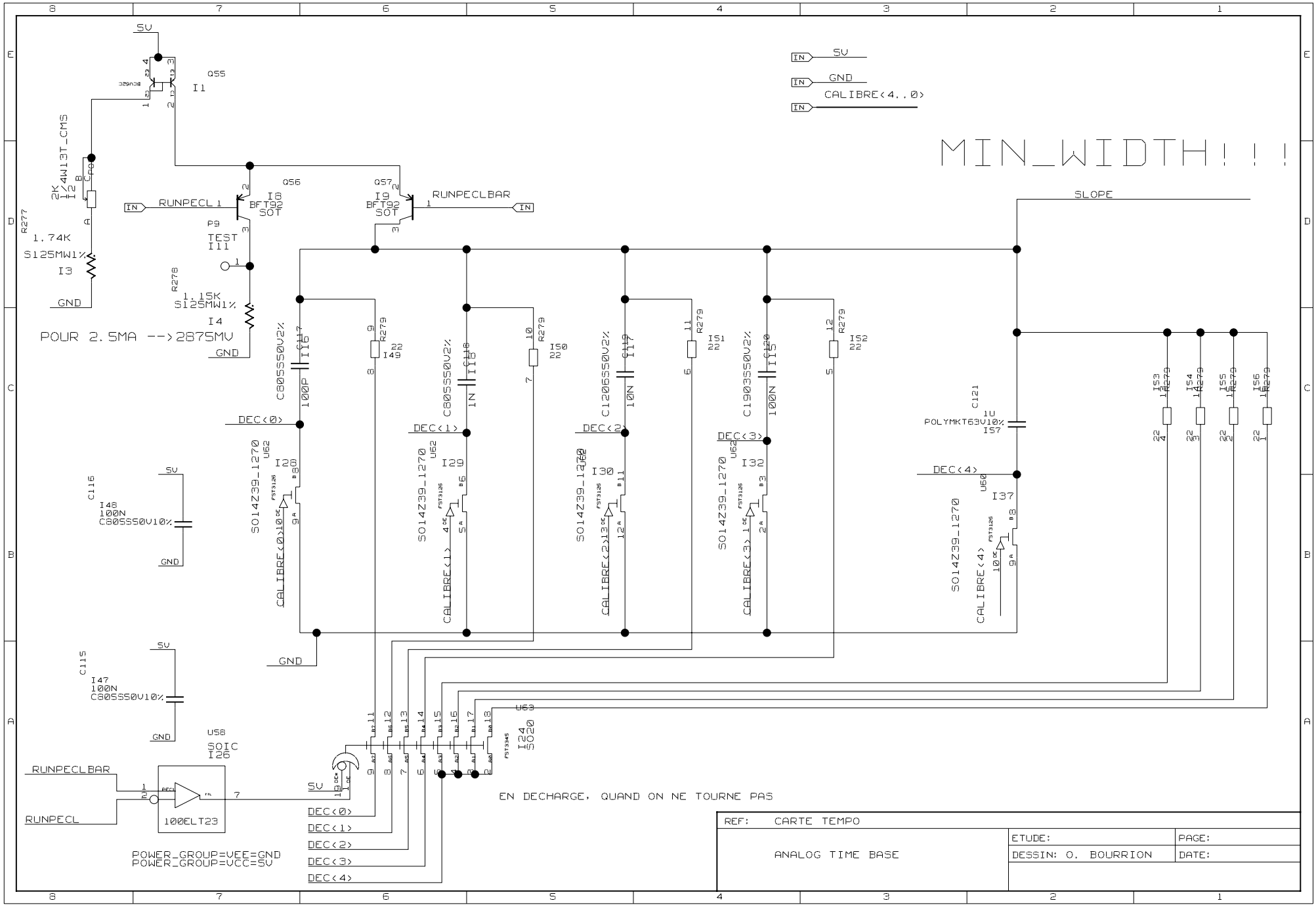
REF: PECL TO NIM

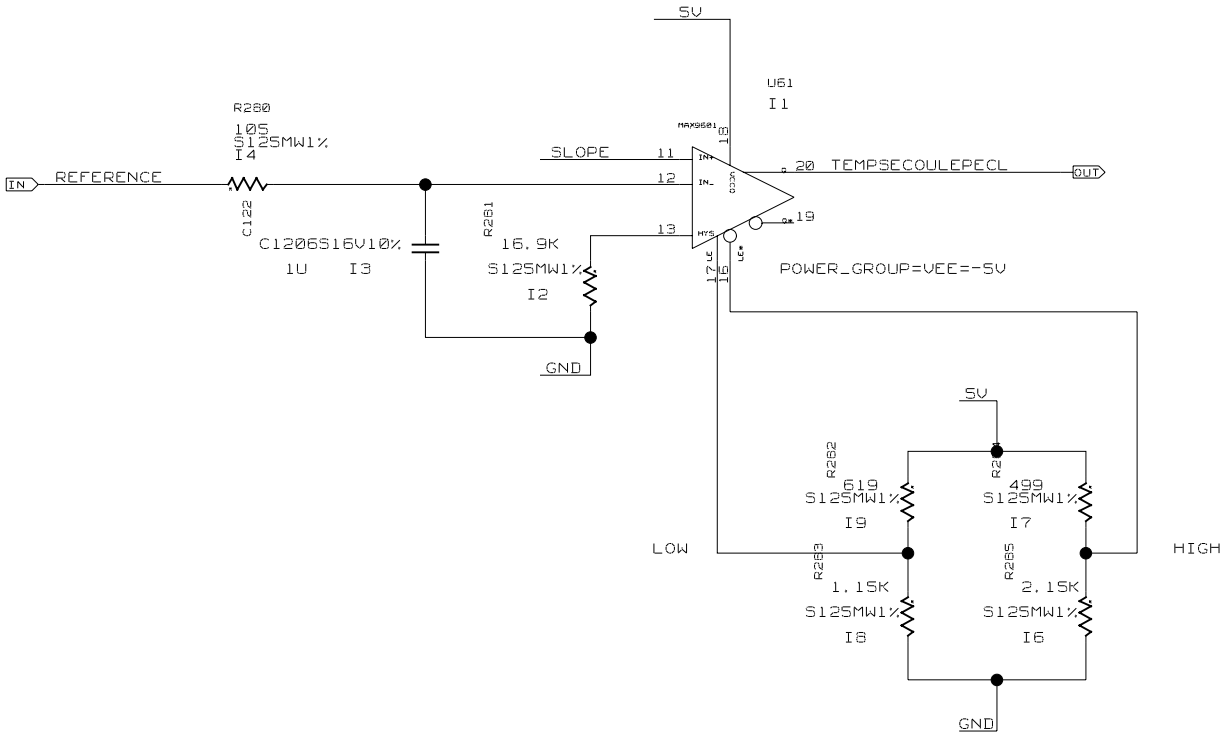
	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!!
5V
GND
CALIBRE<4..0>
IN
SLOPE
RUNPECL
RUNPECLBAR
Q52
Q53
Q54
BFT92
SOT
P8
TEST
R268
1.74K
S125MW1%
R269
1.15K
S125MW1%
POUR 2.5MA -->2875MV
C805S50V2%
100P
1N
10N
100N
R270
DEC<0>
DEC<1>
DEC<2>
DEC<3>
DEC<4>
SO14Z39_1270
U57
U60
CALIBRE<0>
CALIBRE<1>
CALIBRE<2>
CALIBRE<3>
CALIBRE<4>
C113
1U
POLYMKT63V10%
C106
C107
100N
C805S50V10%
U58
SOIC
100ELT23
U59
SO20
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:

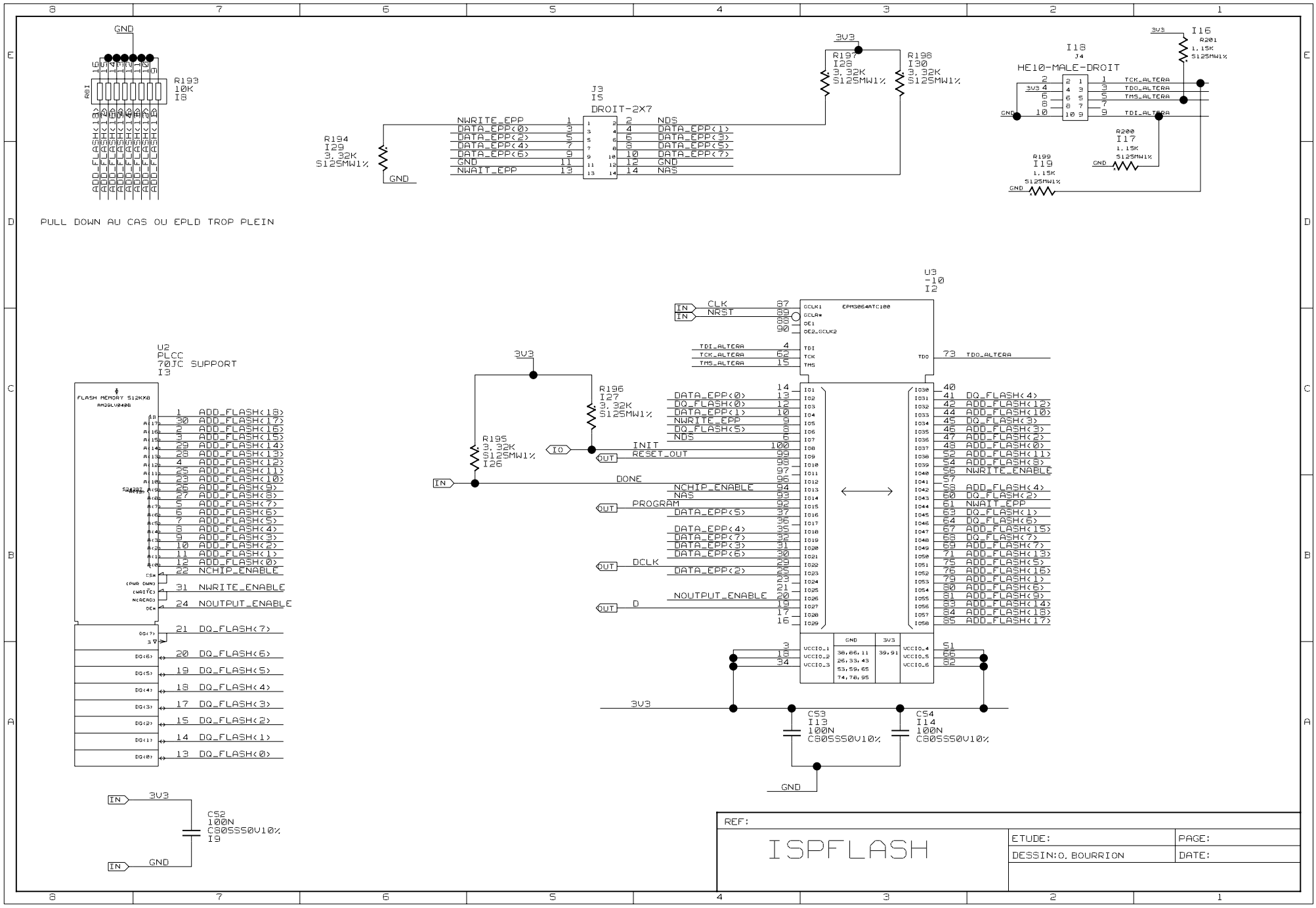


REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:


MIN_WIDTH!!
SLOPE
RUNPECL
RUNPECLBAR
CALIBRE<4..0>
POUR 2.5MA -->2875MV
EN DECHARGE, QUAND ON NE TOURNE PAS
POWER_GROUP=VEE=GND
POWER_GROUP=VCC=5V
100ELT23
SOIC
SO14Z39_1270
REF: CARTE TEMPO
ANALOG TIME BASE
ETUDE:
PAGE:
DESSIN: O. BOURRION
DATE:



REF: CARTE TEMPO		
ANALOG TIME BASE	ETUDE:	PAGE:
	DESSIN: O. BOURRION	DATE:



REF:		
ISPFLASH	ETUDE:	PAGE:
	DESSIN:O. BOURRION	DATE: